



TITLE:

デバイス特性推定に基づく集積回路の適応型テストに関する研究(Dissertation_全文)

AUTHOR(S):

新谷, 道広

CITATION:

新谷, 道広. デバイス特性推定に基づく集積回路の適応型テストに関する研究. 京都大学, 2014, 博士(情報学)

ISSUE DATE:

2014-09-24

URL:

<https://doi.org/10.14989/doctor.k18624>

RIGHT:

許諾条件により本文は2014/11/06に公開

デバイス特性推定に基づく
集積回路の適応型テストに関する
研究

新谷 道広

概要

半導体製造技術の進歩により LSI の大規模化，高性能化が進んでいる．この反面，トランジスタのしきい値電圧やドレイン電流等の特性がばらつく特性ばらつきが問題となっている．LSI が設計仕様どおりに正しく製造されていることを確認する工程をテストという．特性ばらつきの影響で製品仕様を満たさなくなるパラメトリック故障の増大及びテストにおける良否判定基準の設定の困難化が懸念されており，特性ばらつきを考慮し，これらの課題を解決する LSI テスト手法の確立が，今後ならびに将来の LSI 設計に強く求められている．

本論文では，微細プロセスにおける LSI テストのテスト品質の向上とテストに要するコストを削減することを目的とし，特性ばらつき推定に基づく適応型テスト手法を提案する．従来の LSI テストでは，歩留まりの習熟度，特性ばらつき値によらず，全チップに対し同一テストを実施する固定的なテストが主流である．これに対し，提案手法では，テスト前にチップ毎に特性パラメータを推定し，推定結果に基づいて最適なテストを選択してテストを効率化する．本論文では，適応型テストの応用として，従来からある IDDQ テストとパス遅延故障テストを対象とする．IDDQ テストは静止時の電源電流を測定するテストで，パス遅延故障テストは LSI 上のパスを対象としたテストである．両テスト手法とも LSI の代表的な製品規格である動作周波数と消費電力を保証する重要なテストである．チップ毎に，しきい値電圧などの特性パラメータを推定し，期待される回路性能の範囲を計算する．この計算結果を元に，テスト時に，IDDQ テストにおける良否判定基準の決定，パス遅延故障テストにおけるテストすべきパスの変更を行う．従来の固定的なテストと比べて，テストの品質の向上，テストにかかるコストの削減が可能となる．

まず，提案する適応型テストの基盤であり必要不可欠な特性ばらつき推定手法を提案する．特性パラメータの推定精度は，後に提案する適応型テストの効果を決定する．従来の推定手法は，リングオシレータなどの特別な回路を用いて特性パラメータを推定していた．ここで提案する推定手法は，テス

ト時に得られる情報を用いて特性パラメータを推定しており、推定用回路は不要である。本論文では、2種類の推定手法を提案する。1つはIDDQテストにおける測定結果を用いた手法であり、もう1つは、最大動作周波数テストにおける測定結果を用いた手法である。両手法とも、ベイズ推定、最尤推定法を用いて特性パラメータを推定する。両手法を同時に適用することで、推定結果を検証しつつ適用することができ、パラメータ推定の高信頼化に繋がる。計算機実験にて、トランジスタのしきい値電圧を高精度に推定できることを示す。

続いて、特性パラメータ推定結果を用いて、チップ毎にIDDQテストの良否判定基準を決定する手法を提案する。特性ばらつきの増加によりチップのリーク電流のばらつき量が増加し、IDDQテストにおける良否判定基準の設定が困難になっている。その結果、不良チップの見逃し、歩留まり損失が増加し、テスト品質の低下が問題となっている。本手法では、上記で提案したIDDQテスト測定結果を用いた特性パラメータ推定手法に基づき、チップ毎のリーク電流の期待される範囲を計算する。これを用いて良否判定基準を決定する。このように、提案手法は、チップ毎に最適な判定基準を適応的に設定する。計算機実験では、従来のIDDQテスト手法と比べて、故障見逃しと歩留まり損失が低減しテスト品質を向上できることを示す。

最後に、特性パラメータ推定結果を用いた適応型パス遅延故障テストを提案する。特性ばらつきによるトランジスタの遅延時間が変動することで、パス遅延値が製品規格を超えるパラメトリック故障が増加することが問題となっている。パラメトリック故障は特性ばらつきに起因して発生するため、発生するパスがチップ毎に異なる。従来のパス遅延故障テスト手法は、特性ばらつき値によらず、いずれのチップに対しても同一パスをテストするため非効率である。本手法では、特性パラメータ推定結果に応じて、テストするパスを変更する適応型パス遅延故障テストを提案する。同時に、パラメトリック故障に対する故障検出率を提案し、パラメトリック故障に対するテスト品質の定量的な評価を可能にする。計算機実験において、従来の固定的なパス遅延故障テストと比べて、テスト品質を低下させることなくテストコストを削減できることを示す。

以上の提案を通じ、本研究では、特性ばらつき推定に基づく適応型テストがLSIテストにおけるテスト品質を向上し、テストコストを削減できることを示す。特性ばらつき推定手法では、トランジスタのしきい値電圧を5mV以内の精度で推定できる。推定した特性パラメータを用いた適応型IDDQテストにおいては、従来のIDDQテスト手法と比べてテスト品質が14倍向

上できることを示す．最後に，適応型パス遅延故障では，従来のパス遅延故障テストと同等のテスト品質を保持しつつ，テストコストを 10 分の 1 以下に削減できることを示す．

目次

概要	i
第1章 序論	1
1.1 背景	1
1.2 LSIのテスト	3
1.2.1 LSIテストの基本	3
1.2.2 LSIテストの手法分類	3
1.2.3 テスト品質とテストコスト	4
1.3 微細プロセスにおけるLSIテストの課題	5
1.4 従来研究	8
1.4.1 IDDQ電流テスト	8
1.4.2 パス遅延故障テスト	10
1.5 提案する適応型テストの基本概念	12
1.6 論文の構成	14
1.6.1 測定に基づく大域ばらつき推定	15
1.6.2 特性推定に基づく適応型IDDQテスト良否判定基準 決定	15
1.6.3 特性推定に基づく適応型パス遅延故障テスト	16
第2章 測定に基づく大域ばらつき推定	17
2.1 はじめに	17
2.2 IDDQシグネチャを用いたデバイスパラメータ推定手法	18
2.2.1 IDDQシグネチャを用いたデバイスパラメータ推定手法	19
2.2.2 IDDQシグネチャを用いたデバイスパラメータ推定実験	26
2.2.3 まとめ	35
2.3 最大動作周波数テストの枠組みを用いた手法	37
2.3.1 最大動作周波数テスト	37
2.3.2 最大動作周波数テストの枠組みを用いたデバイスパラ メータ推定	37

2.3.3	F_{\max} シグネチャを用いたデバイスパラメータ推定実験	45
2.3.4	まとめ	50
2.4	まとめ	51
第3章	特性推定に基づく適応型 IDDQ テスト良否判定基準決定	53
3.1	はじめに	53
3.2	2 段階 IDDQ テスト手法	54
3.2.1	基本概念	54
3.2.2	クラスタリング方式フィルタ	55
3.2.3	デバイスパラメータ推定に基づく IDDQ テスト良否判定基準決定	57
3.3	シミュレーション実験	63
3.3.1	実験準備	64
3.3.2	従来手法との比較結果	65
3.3.3	リーク故障サイズ毎の検出能力評価の結果	71
3.4	まとめ	72
第4章	特性推定に基づく適応型パス遅延故障テスト	73
4.1	はじめに	73
4.2	適応型パス遅延故障テストフロー	75
4.2.1	設計フェーズ	75
4.2.2	テストフェーズ	79
4.3	パラメトリック故障検出率	79
4.3.1	パラメトリック故障検出率の基本概念	80
4.3.2	適応型テストへの拡張	82
4.4	シミュレーション実験	83
4.4.1	理想クラスタ時の適応型テスト	83
4.4.2	パスクラスタリングを用いた適応型テスト	88
4.5	まとめ	93
第5章	結論	95
5.1	研究成果のまとめ	95
5.2	今後の展望	97
	参考文献	99
	付録	109

目 次	vii
図 目 次	109
表 目 次	113
謝 辞	117
著者による発表論文	119

第1章

序論

1.1 背景

近年の情報社会基盤において、LSI（Large Scale Integrated Circuit）は重要な役割を担っている。我々の生活をより安心、安全、便利にする様々なサービスが提供され、今や無くてはならないものとなっている。LSIの普及に伴い、その信頼性は非常に重要な問題となっており、正しく動作しないLSIの存在は、それを構成するシステムのサービス停止、中断にとどまらず、社会的に大きな影響を与えることにもなりかねない。

LSIを構成する回路要素の物理的欠陥の影響、あるいは回路特性のばらつきによってLSIの製品仕様を満たさなくなること、LSIの出力値が正常な値を示さない状態を故障と呼ぶ[1]。故障の無いLSIを出荷するためには、製造後のテストが必要不可欠である。故障を含むLSIが出荷された場合、そのLSIが搭載されるシステムの信頼性は損なわれる。よって、LSIテストはLSIに品質という付加価値を与える工程であると言える。また、テスト結果を製造プロセスへフィードバックすることで、早期の歩留まり改善に繋げることも重要である。

LSIの製造プロセス技術の一層の進展により、我々の社会はLSIの大規模化、高性能化、低価格化の恩恵を受けている。その反面、微細化により特性ばらつきの問題が深刻化しており、製造技術の改善により特性ばらつきの絶対量は低減されてきているものの、特性ばらつきが回路特性に与える影響は相対的に増大している。ここで、特性ばらつきとは、設計上同じレイアウトで同じサイズのトランジスタであっても、製造された素子のしきい値電圧やチャンネル長などが個々のトランジスタ毎に異なる現象である。特に近年の微細プロセスにおいては、離散不純物揺らぎによるしきい値電圧ばらつきと、ゲートパターンエッジ部の微小な粗さの変化であるラインエッジラフネスによるチャンネル長ばらつきが深刻化している。このように、トランジス

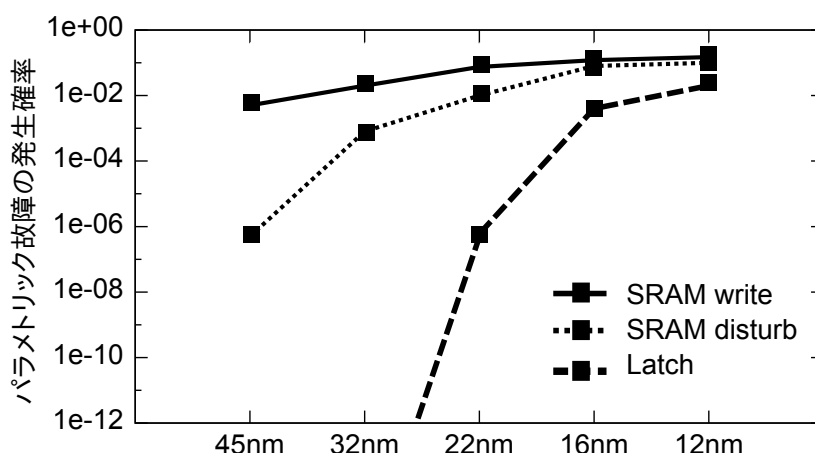


図 1.1: ITRS ロードマップによるパラメトリック故障発生確率予測 [2]

タの特性パラメータがばらつくことで、回路内の信号伝達経路であるパスの遅延値の変動やリーク電流値が増加し、今まで以上に LSI テストが困難となっている。例えば、微細プロセスにおいては、パラメトリック故障と呼ばれる特性ばらつきに起因する故障が多発することが報告されている [3–5]。ITRS (International Technology Roadmap for Semiconductors) が発表した 2011 年度版半導体ロードマップによると、プロセステクノロジー世代が進む毎に、パラメトリック故障の発生確率が上昇することを予想している [2]。図 1.1 に、プロセステクノロジー世代毎のパラメトリック故障発生確率を示す。図 1.1 では、ラッチ及び SRAM (Static Random Access Memory) におけるパラメトリック故障発生確率を示す。SRAM は揮発性メモリの 1 つであり、面積増加を低減するために最小寸法のトランジスタで構成されるため、特性ばらつきの影響を大きく受ける。ここでは、SRAM の書き込み故障と、保持データが破壊されるディスターブ故障を示す。従来プロセスにおいては、パラメトリック故障を対象としたテストは行われておらず、これらの故障をテストコストを増加させることなくテストする技術が必要となる。また、静止電源電流を測定する IDDQ テストにおいては、特性ばらつきの増大に伴うリーク電流の増加により、故障の影響と特性ばらつきの影響の区別が困難になっている。従来の LSI テストでは、特性ばらつきの影響を考慮していないため、IDDQ テストにおける良否判定基準（良品と不良品の判定しきい値）の決定が困難になり、歩留まり損失（良品の不良判定）や故障見逃し（不良品の見逃し）が増加している。こうした課題を解決する特性ばらつきを考慮

した LSI テスト技術の開発が強く望まれている。

1.2 LSI のテスト

1.2.1 LSI テストの基本

製造された LSI 内の故障を検出するためにテストパターンを LSI に入力して出力応答を観測し、予め計算した期待値（回路が正常に動作している時に出力する値）と比較して故障の有無を判定することをテストと言う [1]。出力応答と期待値が一致、あるいは良否判定の基準範囲に収まっていれば良品でありパスとして判定され、不一致、基準範囲外であれば不良でありフェイルとして判定される。故障検出のために入力されるテストパターンとその期待値は、設計者により作成される。一般的にテストは、自動テスト装置（ATE: Automatic Test Equipment）を用いて行われる。

1.2.2 LSI テストの手法分類

歴史的な LSI テスト手法の変遷は、次の 3 つに分類できる [6,7]。

機能テスト（Functional test）：1960 年代における LSI テストの主流となる手法で、LSI が実動作で使われる機能で正しく動作することをテストする。設計時に用いられた検証用シミュレーションデータからテストパターンと期待値を作成し、これを用いてテストを行う。しかし、検証用データは膨大になる傾向があり、テストパタンの作成工数が大きくなるという課題がある [8]。

構造化テスト（Structural test）：仕様と設計は正しいと仮定し、論理ゲートの機能とその接続が通りであるかという観点から回路をテストする。各々のゲートが正しく動いていることをテストすれば、全体としての機能的なテストが行われていなくても良い、という考え方に基づく。通常、構造化テストではテスト容易化設計（DFT: Design for testability）が行われている。代表的なテスト容易化設計手法にスキャン設計がある。スキャン設計用の専用回路（スキャン回路）が、1970 年代より多くの LSI に搭載されている。スキャン設計では、自動テストパターン生成（ATPG: Auto Test Pattern Generator）ツールを用いてテストパターン及び期待値を生成する。

欠陥ベーステスト (**Defect-based test**) : 欠陥とは、期待値と異なる故障の物理的な要因を指す。構造化テストでは、実際の機能を考慮していないために、必ずしも完全なテストはできない。特に、微細なパス遅延量の増加や微小なリーク電流量の増加は、構造化テストだけでは検出が困難である。そこで、物理欠陥の特性を調べ、LSI が物理欠陥を有すると思われる特性を示すならば、不良と判定する [7]。2000 年代より研究が盛んに行われるようになり、適用事例報告も相次いでいる [6,7,9-11]。

現在、半導体メーカーにおいては構造化テストが主流であり、構造化テストでは検出できない不具合に対しては、機能テストが追加で適用されてきた。近年、LSI の信頼性を高めるためのテストとして、欠陥ベーステストが広く研究されている。本研究では、LSI の高信頼化に不可欠な欠陥ベーステストを対象とする。

1.2.3 テスト品質とテストコスト

量産工程でチップそのものをテスト対象とする場合、テスト対象チップを DUT (Device Under Test) と呼ぶ。LSI テストの目的は、DUT 内の故障を検出することである。実際に故障を検出するためには、テスト方法が適切、かつテストパターンが故障検出に適している必要がある。テストにおいて、故障を検出する能力をテスト品質として考えることができる [12]。DUT 内の故障を検出することができれば、テスト品質は高いと言え、逆に検出できずに故障を見逃す場合は、テスト品質が低いと言える。また、良品とすべき DUT を誤って不良品として判定してしまう歩留まり損失も、テスト品質を低下させる要因である。テスト品質の向上は、LSI テストにおける 1 つの目標である。

一方で、テストコストをいかに削減するか、という目標も存在する。テストコストはチップ単価に含められるため、テストコストを低減できれば、安価にチップを提供することができる。テストコストの内訳として、テスト実行時間、使用する ATE の償却費用、テストパターン生成時間、テスト用に付加される回路の面積がある。一般に、ATE は非常に高価な設備である。テスト時は、ATE を占有するため、テスト実行時間、つまりテストに入力されるテストパタンの数はテストコストに直結する。また、テスト品質とテストコストにはトレードオフの関係がある。テストパターン数を増やせば増やすほど、テスト品質が向上する可能性があるが、テストコストは増加する。

以上をまとめると、LSI テストでは次の 2 点を達成することが重要となる。

1. テスト品質を最大化
2. テスト品質を低下させずにテストコストを最小化

1.3 微細プロセスにおける LSI テストの課題

微細プロセスにおいて特性ばらつきが増大し、LSI テスト品質の低下、及びテストコストの増加が顕著になっている。本節では、微細プロセスにおける LSI テストの課題についてまとめる。特性ばらつきの増大は、回路特性であるリーク電流値とパス遅延値に大きな影響を与える [13]。以下に、リーク電流とパス遅延の増加がテストに与える影響について述べる。

電流値を測定するテストに IDDQ テストがある。IDDQ テストは、CMOS トランジスタの特徴である信号値の静止時にはほとんど電流が流れない特徴を利用し、静止時の電源電流値を測定すること (IDDQ 測定) で、チップ内の欠陥の有無を判定するテスト手法である [14]。特性ばらつきの増大により、IDDQ テストの良否判定基準の適切な設定が困難になっており、歩留まり損失および故障見逃し増加の要因となっている [15]。

IDDQ テストのテスト品質低下の要因は、特性ばらつきの増大である。リーク電流値は、サブスレッショルドリーク電流、ゲートリーク電流、接合リーク電流から構成される。ここでは、IDDQ 電流の大部分をサブスレッショルドリーク電流で占めると仮定する [16]。サブスレッショルドリーク電流 I は、次のように表現される [17]。

$$I = I_0 \exp\left(\frac{V_{gs} - V_{th}}{\eta V_T}\right) \left(1 - \exp\left(\frac{-V_{ds}}{V_T}\right)\right) \quad (1.1)$$

ここで、 $I_0 = \mu_0 C_{ox} (W_{eff}/L_{eff}) V_T^2 \exp(1.8)$ である。 $V_T = kT/q$ であり、 k はボルツマン定数、 μ_0 は電荷キャリアの移動度である。 C_{ox} は単位面積当たりのゲート酸化膜容量を表す。 T は絶対温度、 q は電子の電荷、 η は指数係数でありプロセステクノロジーに依存する。 V_{gs} と V_{ds} はゲート-ソース間とドレイン-ソース間の電圧である。 V_{th} はしきい値電圧で、 W_{eff} と L_{eff} は有効チャネル長とチャネル幅である。式 (1.1) から、サブスレッショルドリーク電流値 I はしきい値電圧 V_{th} に応じて指数関数的に変化することが分かる。よって、特性パラメータであるしきい値電圧がばらついた時、リーク電流値にも大きな影響を与える。

IDDQ テスト良否判定基準の適切な決定が困難になっていることを一般化した例を図 1.2 に示す。ここでは、IDDQ 電流を階級とした良品チップと

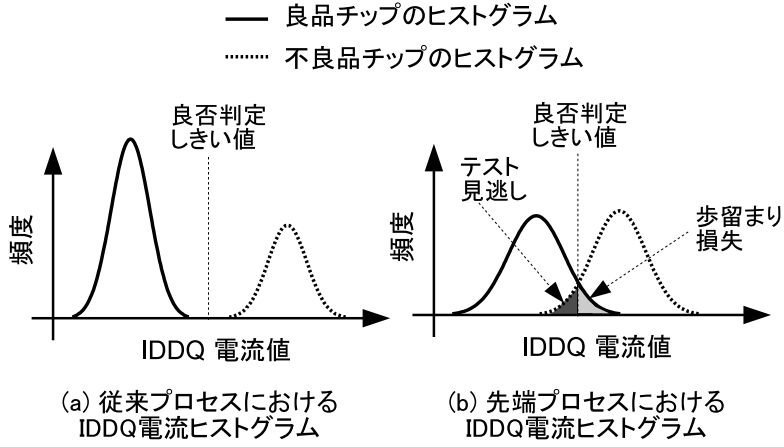


図 1.2: IDDQ テスト良否判定基準決定の困難化

不良品チップのヒストグラムを示す。従来プロセスにおいては、ヒストグラムは図 1.2(a) に示すような形状をしていた。図 1.2(a) の右側の分布は、明らかに異常値を持つチップ群であり、良否判定しきい値の設定が容易であった。ところが、先端プロセスでは特性ばらつきの増大により IDDQ 電流値が増加し、図 1.2(b) に示すように、良品チップの分布と不良品チップの分布が重なる。リーク電流の増加要因は、製造時に混入する粒子によるショートやオープンなどの物理欠陥も含み、微小なリーク電流量の増加を生ずる物理欠陥であれば、特性ばらつきによる増加リーク電流と区別する事は困難である。このように、IDDQ テストにおける故障見逃しと歩留まり損失が増加し、テスト品質が低下している。

続いて、特性ばらつきによるパス遅延値の変動が LSI テストに与える影響を考える。パス遅延値もリーク電流と同様に、しきい値電圧がばらついた時に大きく変動する。パス遅延値は次のような線形式で表現される [18]。

$$d = \mu_d + \sum_{i=1}^n S_{p_i} \Delta p_i + N(0, \sigma_{rnd}^2) \quad (1.2)$$

ここで、 d はパスの遅延値、 μ_d は d の平均値、 Δp_i は i 番目の特性ばらつきで、 S_{p_i} は Δp_i に対するパス遅延感度である。 $N(0, \sigma_{rnd}^2)$ は局所ばらつきによるパス遅延の変動を表し、平均値が 0、標準偏差が σ_{rnd} の正規分布である。局所ばらつきとは、主に特性ばらつきに占めるチップ内のばらつきを指す [19]。式 (1.2) から、リーク電流と同様、特性ばらつきの増減に応じてパス遅延値が変化することが分かる。

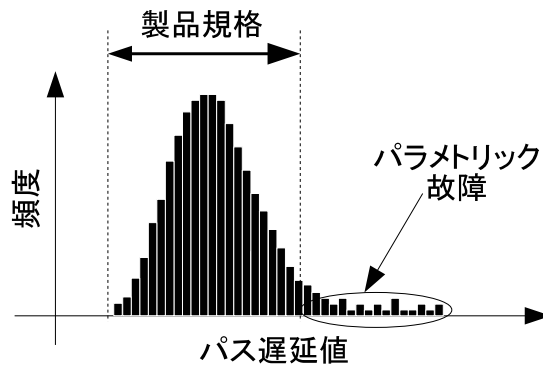


図 1.3: パラメトリック故障の概念図

これに加えて LSI の高速化に伴う設計マージンの削減され、特性ばらつきに起因して故障を引き起こすパラメトリック故障と呼ばれる現象が問題になっている [3–5]. 図 1.3 にパラメトリック故障の概念図を示す. 図 1.3 では、回路のパス遅延値を階級としたチップ毎のヒストグラムを示している. パラメトリック故障は高速な LSI のクリティカルパス上で発生しやすい. 図に示すように、特性ばらつきによって回路のパス遅延値が増大し、製品規格を外れることで発生し LSI の誤動作を引き起こす. 特性パラメータは製造上よく管理されていることが前提となっているが、微細プロセスにおいて、特にプロセス開発初期においては困難である. 統計的静的タイミング解析 (SSTA: Statistical Static Timing Analysis) ツールは、特性ばらつきを統計量として扱い、正確にパス遅延値を予測できる [18]. SSTA を用いた時、パス遅延値の平均値と標準偏差が μ , σ と予測され、 3σ 設計 ($\mu + 3\sigma$) が行われた場合を考える. これは、約 0.13 % の割合で製品規格を満たさないチップが製造されることを意味し、これらのチップが市場へ出荷されることを防ぐために、パラメトリック故障に対するテストが必要である [20].

パラメトリック故障をテストするための手法として、パス遅延故障テストが適している [21,22]. パス遅延故障用のテストパターンは、パス遅延故障モデルを対象として生成される [23]. 図 1.4 にパス遅延故障モデルの概念図を示す. 図 1.4 に示すように、パス遅延故障モデルは、フリップフロップ間を結ぶパス上を対象としており、パス上の回路素子における信号伝搬遅延が蓄積し、製品規格、つまりクロック周波数が目標に届かない故障をモデル化している. よって、パス遅延故障モデルは遅延要素が回路内に小さく分散している場合に有効である [24]. パラメトリック故障は、特性ばらつきによって

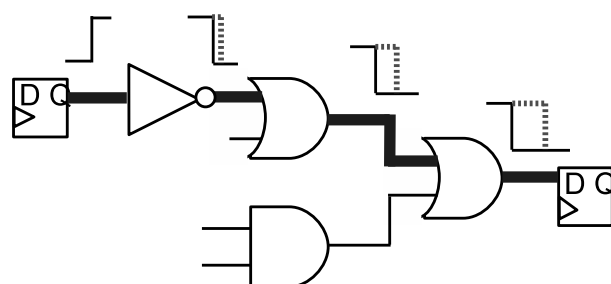


図 1.4: パス遅延故障モデル

パス上の回路素子の遅延値が増加することで発生するため，パス遅延故障モデルはこの現象を良く表現している．ただし，回路上のパスは，回路規模に応じて指数関数的に増大するため，全パスに対して適用することは非現実的である．どのパスを対象とすれば良いか，という明確な指針がないことが課題である [25]．

従来の回路特性のばらつきは，製品規格に対して相対的に小さく，製品規格との設計マージンは十分確保されていた．さらに，上記に上げたパス遅延故障テスト適用の課題もあり，車載向け LSI などの高品質が要求される LSI 以外でのパス遅延故障テストの適用事例は多くない．よって，テストコストを過剰に増大させることなく，パラメトリック故障に対するテスト品質を向上させることが重要である．

以上のように，特性ばらつきが増大することで，LSI テストにおけるテスト品質の低下及びテストコストの増加が懸念される．

1.4 従来研究

ここでは，1.3 節で述べた IDDQ 電流テストとパス遅延故障テストについて述べる．両テストとも，消費電力規格と速度性能規格を保証するために重要である．

1.4.1 IDDQ 電流テスト

IDDQ 電流シグネチャを用いた良否判定基準設定の容易化手法が多数提案されている [26–29]．テストパターン毎に IDDQ 電流値を測定した IDDQ 電流

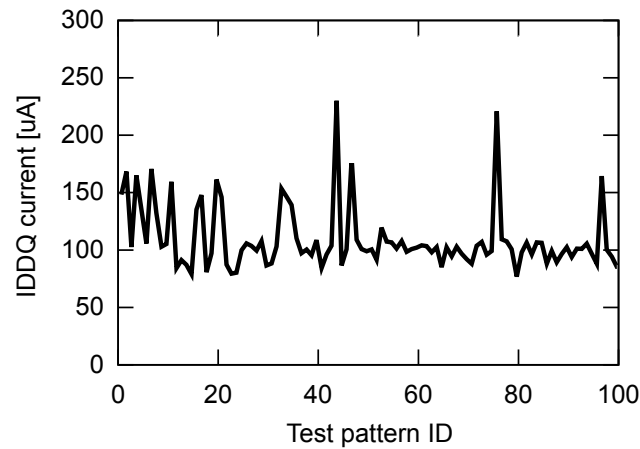


図 1.5: IDDQ 電流シグネチャの例

の系列を IDDQ シグネチャと呼ぶ [26]. IDDQ 電流シグネチャの例を図 1.5 に示す. IDDQ テストでは, スキャン設計を用いてテストパターンが印加される. この例では, 100 パターン印加した場合を示している. テストパターン間の IDDQ 値の差は回路の状態差, もしくは欠陥起因の増加電流による変調を意味する. 文献 [26–28] では, デルタ IDDQ 手法が提案されている. 測定した IDDQ シグネチャ内の IDDQ 電流値を昇順にソートし, 隣接する IDDQ 電流値の差分を比較している. この電流変化がある一定値を超えた場合に不良品として判定される. 文献 [29] では, カレントレシオ手法が提案されている. 本手法では, IDDQ シグネチャ内のテストパターン数に対する最大電流値と最小電流値の傾きがあるしきい値を超えると不良品として判定する. 文献 [26–29] で提案されている手法は, テストパターン毎の回路状態間でリーク電流値の差が小さいことに着目しており, 特性ばらつきを考慮していない.

文献 [30–32] に, 特性ばらつきを考慮した統計的リーク電流解析手法が提案されている. これらの手法を用いてリーク電流の上限値を統計的に求めることで, IDDQ テストの良否判定に用いることもできるが, これらの手法は IDDQ テストを考慮していない.

特性ばらつきを考慮した IDDQ テスト手法に, NNR (Neighbor Nearest Residual) 法がある [33, 34]. NNR 法は, IDDQ シグネチャを用いずに実測値と推定値の残差を用いる. 実測値は, IDDQ テスト時に得られた IDDQ 測定値である. 推定値は, ウェハ上の DUT 周辺チップの実測値の平均である. DUT に故障が無ければ, 残差は非常に小さくなり, 故障があれば残差は大きくなる. NNR 法は, DUT の IDDQ 電流値と周辺チップの IDDQ 電

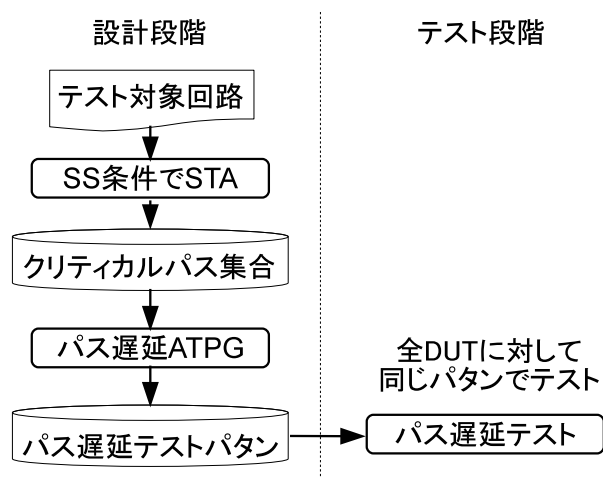


図 1.6: 一般的なパス遅延故障テスト適用フロー

流値は近い数値であることを仮定しており、この仮定が成り立たない場合、NNR 法の有効性は低減する。

式 (1.1) から、IDDQ 電流値は特性ばらつきに応じて大きく異なることを示した。特性ばらつきはチップ毎に異なるため、IDDQ 電流値はチップ毎に異なる [30]。チップ毎の特性ばらつきを知ることで、IDDQ 電流値を推定することができ、良否判定基準を精度良く設定することができる。

1.4.2 パス遅延故障テスト

パス遅延故障テストは、微小な遅延故障を検出できる高品質テスト手法として期待されており、長く研究の対象となっている [35–42]。文献 [35,36] では、テスト対象パスから機能的に活性化できないパスを識別して、パス遅延故障テストの効率を向上させている。文献 [37,38,41] では、SSTA ツールを用いてテスト対象パスを選択する手法を提案している。文献 [39] では、パス遅延に対する遅延故障、ノイズ、特性ばらつきの感度を計算し、この感度を基にテスト対象パスを選択する手法を提案している。文献 [42] では、機能検証用パターンを用いて、テスト対象パスを選別する手法を提案している。文献 [40] では、パス遅延故障テストの ATPG の問題を SAT (充足可能性問題) に帰着させて解くことで、ATPG のテストパターン生成能力を飛躍的に向上させている。

ここで、パス遅延故障テストの対象パスの選択について考察する。図 1.6

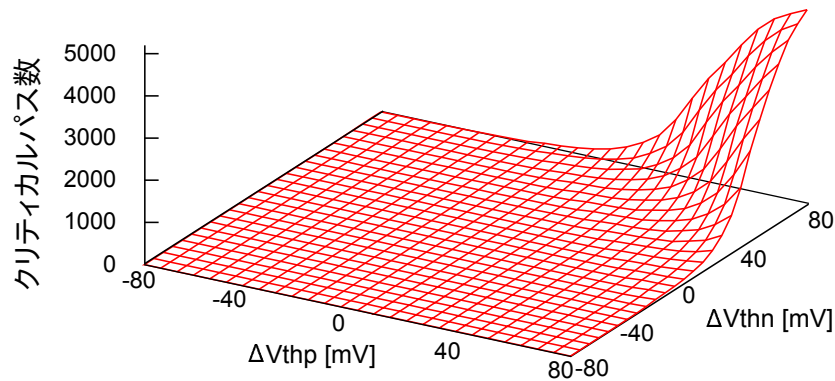


図 1.7: 想定するばらつき空間におけるクリティカルパス数の変化

に、一般的なパス遅延故障テストの適用フローを示す [43]. まず、パス遅延故障テストの対象回路に対して、静的タイミング解析 (STA: Static Timing Analysis) ツールを用いてテスト対象であるクリティカルパスを抽出する. クリティカルパスに対してパス遅延故障テスト用のテストパターンを生成する. 生成したテストパターンを用いてパス遅延故障テストを行う. 一般的に、故障見逃しによるテスト品質低下が発生することがないように、STA では悲観的にパス遅延値を見積もりクリティカルパスを抽出する. すなわち、STA において、nMOS と pMOS のトランジスタのスイッチングスピードがいずれも遅い場合、すなわち Slow-slow (SS) 条件を想定する. 悲観的にパス遅延値を評価しているため、多くのパスがクリティカルパスとして抽出される. テスト対象パスが増えるほど、パス遅延故障テストパターンは増加するためテストコストは増大する. DUT が Fast-fast (FF) 条件に属する場合、これらのテストパタンの大部分は明らかに無駄となる.

図 1.7 に、想定する特性ばらつき空間におけるクリティカルパス数の変化を示す. ここでは、OpenCores ベンチマーク回路 [44] の Ethernet 回路を 65-nm プロセスの標準セルライブラリを用いて設計した例を示す. 特性ばらつき空間を nMOS と pMOS トランジスタのしきい値電圧のばらつき ΔV_{thp} と ΔV_{thn} の 2 次元としている. $(\Delta V_{thn}, \Delta V_{thp}) = (-80 \text{ mV}, 80 \text{ mV})$ の時が FF 条件で、 $(\Delta V_{thn}, \Delta V_{thp}) = (80 \text{ mV}, -80 \text{ mV})$ の時が SS 条件である. ばらつき空間の中心 $(\Delta V_{thn}, \Delta V_{thp}) = (0 \text{ mV}, 0 \text{ mV})$ はノミナル条件である. 図 1.7 から、FF 条件から SS 条件を結ぶ線上において、FF 条件からノミナル条件にかけてはクリティカルパス数は 0 である. 一方で、ノミナル条件から SS 条件にかけては指数関数的にクリティカルパス数が増加している. 既存のパス遅延故障テストにおいては、常に SS 条件を想定してテストパターンを想定して

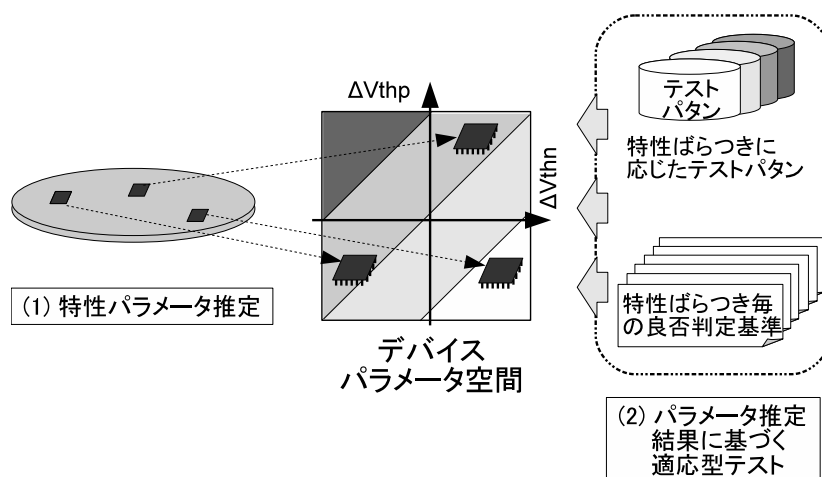


図 1.8: 推定に基づく適応型テストの概念

おり，その時のクリティカルパス数は4,601パスである．製造したチップは，想定した ΔV_{thp} , ΔV_{thn} の空間内のいずれかの数値をとるが，常に，SS 条件である $(\Delta V_{thn}, \Delta V_{thp}) = (80 \text{ mV}, -80 \text{ mV})$ となることはない．製造上ばらつきを小さく抑えるように管理されているため， $(\Delta V_{thn}, \Delta V_{thp}) = (0 \text{ mV}, 0 \text{ mV})$ が最も発生しやすいことが期待される．よって，予め DUT の特性ばらつきが分かれば，対象とするクリティカルパスを絞り込むことができる．

1.5 提案する適応型テストの基本概念

本研究では，LSI テストにおける，テスト品質の向上とテストコストの削減を目的として，特性ばらつき推定に基づく適応型テスト手法を提案する．図 1.8 に，提案する適応型テストの概念図を示す．この適応型テストは，(1) テスト前に DUT 毎に特性ばらつきを推定する，(2) 得られた推定特性ばらつきに基づいてテストの内容を適応的に変更する，点が特徴である．従来から，DUT 毎にテスト内容を変更する適応型テストの考えは適用されているが，テスト技術者の経験に依存している．本研究で提案した適応型テストは，特性ばらつき推定から DUT 毎の回路性能を見積りテスト内容を変更する．

本論文では，適用例として，提案する適応型テストをパス遅延テストと IDDQ テストに応用する．IDDQ テストとパス遅延故障テストは，LSI の重要な製品規格である消費電力規格と動作周波数規格に関わるテスト手法であり重要である．両テストは独立したテストであり，量産テストにおいては両

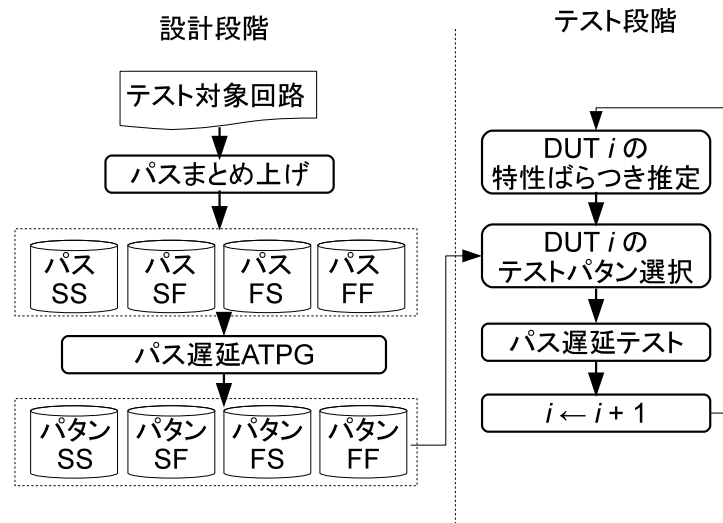


図 1.9: 適応型パス遅延故障テストの概念図

方適用される。推定した特性ばらつき値から、DUT 毎のパス遅延値，リーク電流値の上限値が推定できる。これらを用いて，適応的にテストに用いるテストパタンの変更，IDDQ テストにおける良否判定基準の変更を行う。提案手法を用いることで，LSI テストにおけるテスト品質の向上とテストコストの削減が可能となる。

図 1.9 に，適応型テストのパス遅延故障テストへの適用例を示す。従来手法である図 1.6 では，SS 条件を想定したテストパタンが全量産 DUT に対して適用されてきた。提案手法を適用することで，特性ばらつきに応じて，テストパタンを変更できる。図 1.9 では，特性ばらつきに応じて，特性ばらつきが FF, SF, FS, SS の 4 条件の時のパス集合にまとめ上げている。これらのパス集合に対して，パス遅延 ATPG を行う。テスト時には，パス遅延テストを行う前に，DUT 毎の特性ばらつきを推定し，その DUT に最も適したテストパタンを選択して，パス遅延故障テストを行う。従来の悲観的なテストから，より最適なテストを行うことができるため，テスト品質を低下させることなく，テストコストを削減することができる。

ここで，本論文で扱う特性ばらつきについて整理する。特性ばらつきはその性質から次の 3 種に分類できる [19]。

- 大域ばらつき
- ランダムな局所ばらつき
- システムティックな局所ばらつき

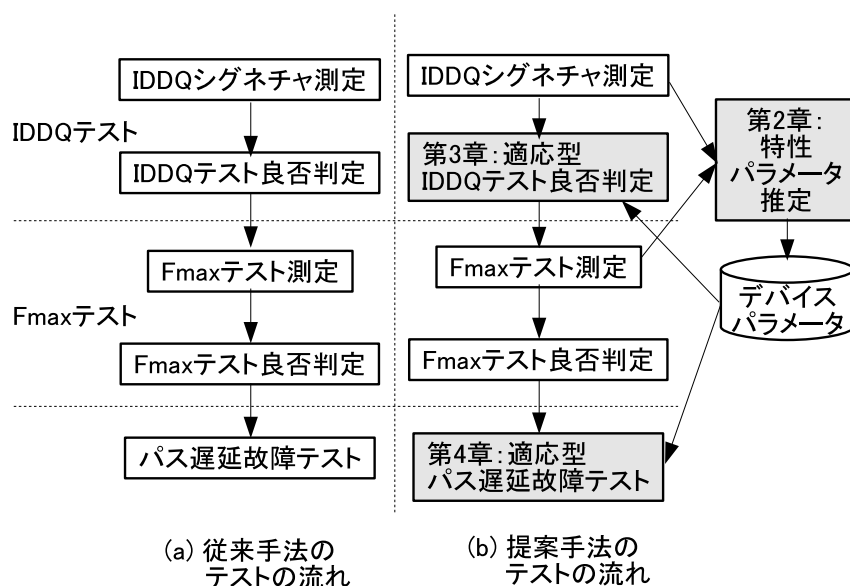


図 1.10: 本論文で想定するテストの流れと論文の構成

大域ばらつきは、ウェハ間、ロット間でのばらつきを含むチップ間の回路特性ばらつきを指す。ウェハ間、ロット間は、製造条件の差異に起因する。同ウェハ上のチップ間ばらつきは、製造時の熱分布などに起因する成分で、隣接するチップ間は連続な曲面で表すことができる。局所ばらつきは、その振る舞いから、ランダム成分とシステムティック成分に分類される。ランダムな局所ばらつきは、主にチップ内のばらつきで、製造時のガウス雑音に起因する。システムティックな局所ばらつきは、統計的にはばらつかないが、チップ上の場所に応じてモデル化することができる成分である。システムティックな局所ばらつきは、レイアウト上のパターン密度、隣接パターン間の距離、CMP（Chemical Mechanical Polishing）ばらつきに起因する。1.4 節で述べたように、従来のテスト手法は特性ばらつきを考慮していない。本研究では、特性ばらつきにおける大域ばらつき成分を推定対象のデバイスパラメータとし、適応型テストに用いる。

1.6 論文の構成

図 1.10 に、本論文で想定するテストの流れと論文の構成を示す。図 1.10(a) に従来のテストにおけるテストの流れを示す。テストは、IDDQ テスト、最

大動作周波数 (F_{\max}) テスト, パス遅延テストが行われることを想定する. F_{\max} テストは, 最大動作周波数値 (F_{\max} 値) を測定して故障の有無を判定するテスト手法である [45]. IDDQ テストと F_{\max} テストは, 測定と測定結果による良否判定の2工程に分かれる. 特性ばらつき推定に基づく適応型テストを IDDQ テストとパス遅延テストに適用することで, 図 1.10(b) のようになる. まず, 第2章では, 適応型テストに向けたデバイスパラメータ推定手法を提案する. デバイスパラメータ推定は, 適応型テストにおいて必要不可欠である. ここでは, テスト時に得られる情報を用いたデバイスパラメータ推定手法を提案する. 続く第3,4章で, 特性ばらつき推定に基づく適応型テストの適用例として, 適応型 IDDQ 良否判定基準決定と推定に基づく適応型パス遅延テストを提案する. 第3,4章における適応型テスト手法の効果は, 第2章のデバイスパラメータ推定の精度に依存する. 第3,4章で, 提案する適応型テストを用いることで, LSI テストにおけるテスト品質が向上しテストコストが削減できることを示す. 最後に, 第5で本研究のまとめと今後の展望について述べる.

1.6.1 測定に基づく大域ばらつき推定

第2章では, 適応型テストのためのデバイスパラメータ推定手法を提案する. 本推定手法は, テスト時に得られる情報を用いて推定する. ここでは IDDQ テスト手法と F_{\max} テスト手法から得られる情報を用いた2種類の推定手法を提案する. 両手法とも, まず, ライブラリとして, 回路の統計的なリーク電流分布, 最大周波数分布を得る. 続いてそのライブラリを用いて, 最後に IDDQ テストによって得られた IDDQ 電流値, あるいは F_{\max} テストから得られた F_{\max} 値に対してベイズ推定, 最尤推定を適用してチップ毎の特性パラメータを推定する. IDDQ テストと F_{\max} テストは独立して行われるテストであるため, 両方の推定手法を適用することができ, 検証確認しながら推定を行い, 高信頼な推定結果を得ることができる. 計算機実験から, 両手法とも特性パラメータを高精度に推定できることを示す.

1.6.2 特性推定に基づく適応型 IDDQ テスト良否判定基準決定

第3章では, IDDQ テストにおける適応的なテスト良否判定基準値の決定方法を提案する. 本手法では, 第2章で提案した IDDQ 測定電流値を用いた推定手法を用いて, IDDQ テストの良否判定良否判定基準を決定する.

チップ毎に推定された特性パラメータを用いることで、リーク電流内の故障起因の電流とばらつき起因の電流を分離できる。テスト毎に適切な良否判定基準を設定し、故障見逃しと歩留まり損失を低減する。仮想ウェハを想定した計算機実験において、従来手法である NNR 法 [33,34] と比較し、故障見逃しと歩留まり損失を低減でき、テスト品質が向上されることを示す。

1.6.3 特性推定に基づく適応型パス遅延故障テスト

第 4 章では、デバイスパラメータ推定結果から、DUT 毎にテストすべきパスを変更する適応型パス遅延故障テストを提案する。適応型パス遅延故障テストでは、設計フェーズで特性パラメータ毎にクリティカルパスの集合であるパスクラスタを得る。全パスクラスタに対してパス遅延故障 ATPG を行い、対応するテストパターンを生成する。テスト時に、DUT の特性パラメータを推定し、推定結果に応じて対応するテストパターンを用いてパス遅延故障テストを行う。また、第 4 章では、パラメトリック故障に対する故障検出率を提案し、パラメトリック故障に対するテスト品質を定量化する。提案する適応型テストを用いることで、テスト品質を保持しつつ、テストコストを削減でき、効率的にパラメトリック故障をテストできることを示す。

第2章

測定に基づく大域ばらつき推定

本章では，第3,4章で提案する適応型テストに必要不可欠となるデバイスパラメータ推定手法を提案する．特性ばらつき推定に基づく適応型テストの効果はデバイスパラメータ推定精度に依存するため，本推定手法は非常に重要である．本章では，LSIテストの測定時に得られる情報を用いたデバイスパラメータ推定手法を提案する．本推定手法では，特性ばらつきの分布をモデル化し，ベイズ推定，最尤推定法に基づいてデバイスパラメータを推定する．本章では，提案する推定手法を用いることで，高精度にデバイスパラメータを推定できることを示す．

2.1 はじめに

デバイスパラメータを簡易的に推定する手法として，リングオシレータの発振周波数を測定する手法が提案されている [46–49]．リングオシレータを事前に回路中に埋め込んでおくことで，ウエハテスト時や製品出荷時，または出荷後にも，簡易的にチップ性能を見積もることができる．リングオシレータがインバータ等の単一の論理セルにより構成されている場合，発振周波数の変化は pMOS, nMOS トランジスタを総合した特性ばらつきや環境変動を示す．文献 [47,49] では，複数の論理セルを組み合わせで大域ばらつきを高精度に推定する．また，文献 [48] では，事前にリングオシレータのパス遅延分布を計算し，リングオシレータの測定結果から最尤推定法を用いて，大域ばらつきのみならず，局所ばらつきまで推定する手法を提案している．文献 [49] また，リングオシレータを用いない手法として，回路内のタイミング余裕のあるパスに付加的に遅延を挿入し，新たなパス遅延とクロック周期との関係からチップ毎の大域ばらつきを推定する手法も提案されている [50]．しかし，これらの手法は全て，パラメータ測定用の追加回路と推定

のための追加測定が必要になる。

本章では、テスト時の情報を用いたデバイスパラメータ推定手法を提案する。提案手法は、推定のために必要な情報をテストから得られるため、追加回路や測定が不要である点が大きな特徴である。本章では、2種類の推定手法を提案する。1つ目の手法は、IDDQテスト測定で得られるIDDQシグネチャを用いる。2つ目の手法は、最大動作周波数(F_{\max})テストの枠組みから得られる測定結果を用いる。両手法とも推定の方法としてベイズ推定、最尤推定法を用いることで高精度にデバイスパラメータを推定できる。IDDQテストと F_{\max} テストは別々に適用されるテストであるため、両推定手法を同時に適用することができる。これにより、推定結果を確認しながら適用でき、推定結果の高信頼化に繋がる。ここで、両推定結果に差異があった場合、後の適応型テストにおける故障見逃しを防ぐため、悲観的なテストが行われる推定結果を用いる。

本章で提案する本推定手法は、テスト時に得られる情報を用いて推定する。良否判定が行われる前のチップは、物理欠陥による故障の影響を含む場合があるため、チップに故障がある場合、特性推定が正しく行われない可能性がある。本節では、テストで良品と判定されたチップのテストの測定情報を用いると仮定し、本章では提案するデバイスパラメータ推定手法の原理を述べる。しかし、適応型テストで、本手法を用いるためには、測定の情報から故障の影響を除外する必要がある。これについては第3章にて述べる。

以下、本章は次のように構成する。2.2節にて、IDDQシグネチャを用いたデバイスパラメータ推定手法を提案し、2.3節にて、 F_{\max} テストの枠組みを用いたデバイスパラメータ推定手法を提案する。2.4節で本章をまとめる。

2.2 IDDQシグネチャを用いたデバイスパラメータ推定手法

本節では、IDDQシグネチャを用いたデバイスパラメータ推定手法を提案する。本推定手法では、まず、ライブラリとして、セル毎の統計的リーク電流の分布を得る。続いて、そのライブラリを用いて、対象回路全体における統計的リーク電流の分布を計算する。最後にIDDQテストによって得られたIDDQシグネチャからベイズの定理を用いてチップ毎のデバイスパラメータ、すなわち大域ばらつきを推定する。IDDQシグネチャは、テストパターン毎にIDDQ値を測定したIDDQ電流の系列である [26]。また、LSI量産

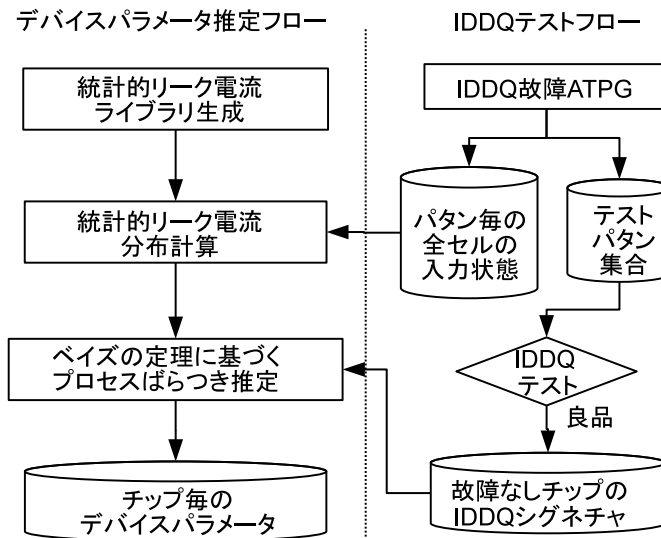


図 2.1: IDDQ シグネチャを用いたデバイスパラメータ推定手法の全体フロー

テストにおいて広く行われている IDDQ テストの測定結果を用いることで、デバイスパラメータ推定のための回路の搭載や追加の測定が不要である。

2.2.1 IDDQ シグネチャを用いたデバイスパラメータ推定手法

本節では、IDDQ シグネチャを用いたデバイスパラメータ推定手法を提案する。図 2.1 に、提案手法の全体フローを示す。本推定手法において必要な情報は、IDDQ テストフローから得ることができるため、特別な回路の追加、テストフローの変更が不要である。IDDQ テストフローにおいて、まずテスト対象回路に対し IDDQ 故障検出のためのテストパターンを自動テストパターン生成（ATPG）ツールにより生成する。同時に、それぞれのテストパターンに対応する対象回路内の状態を得る。チップの製造後、IDDQ テストを実施し、先に生成した IDDQ 故障テストパターンを用いて IDDQ 電流値を測定する。本推定手法では、良品チップの IDDQ シグネチャを用いてデバイスパラメータを推定する。

本推定手法は、次の 3 工程からなる。

1. ゲートレベル統計的リーク電流ライブラリの作成
2. チップレベル統計的リーク電流分布の計算
3. ベイズの定理に基づくデバイスパラメータの推定

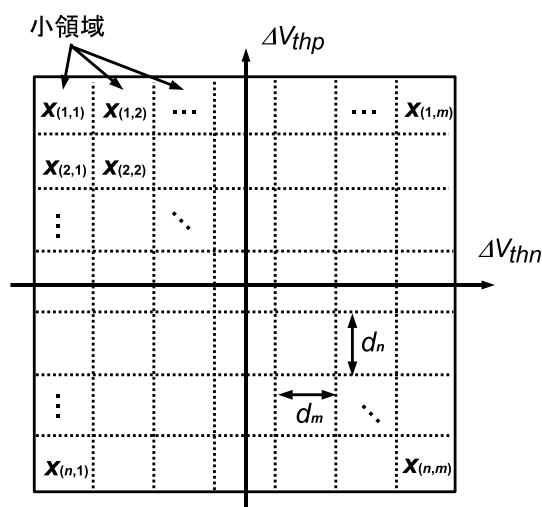


図 2.2: 2 ばらつき変数によるプロセス領域分割例

本推定手法のフローにおいて、まず、ゲートレベル統計的リーク電流ライブラリ（SLL: Statistical Leakage Library）を作成する。ゲートレベル SLL は、与えられた標準ロジックセルセットに対するデバイスパラメータ値に応じた統計的リーク電流分布を記録する。続いて、ゲートレベル SLL と IDDQ 故障 ATPG より得たテストパターン毎の回路状態を用いて、チップレベル SLL を計算する。チップレベル SLL は、デバイスパラメータに応じた回路全体の統計的リーク電流分布を含む。最後に、IDDQ テストから得られた IDDQ 電流シグネチャとチップレベル SLL を用いて、ベイズの定理を適用することで、チップ毎のデバイスパラメータ値を推定する。

ゲートレベル統計的リーク電流ライブラリの作成

本推定手法では、想定するデバイスパラメータの変動空間を、図 2.2 に示すような小領域に分割する。この小領域の大きさは、本推定手法における特性推定の最小分解能である。図 2.2 は、二つのパラメータ nMOS トランジスタと pMOS トランジスタのしきい値電圧の平均値からの変動量 ΔV_{thn} と ΔV_{thp} について 2 次元で分割した例である。パラメータ空間は、 $n \times m$ 個の領域に分割されている。小領域の大きさを表す d_n と d_m は、計算コストと必要なデバイスパラメータ精度のトレードオフの関係にある。すなわち、 d_n と d_m が小さい場合は高精度に推定できるが、計算コストが大きい。以後、 x を用いて小領域を表す。 x は d 次元ベクトルであり、 d はデバイスパラメータ

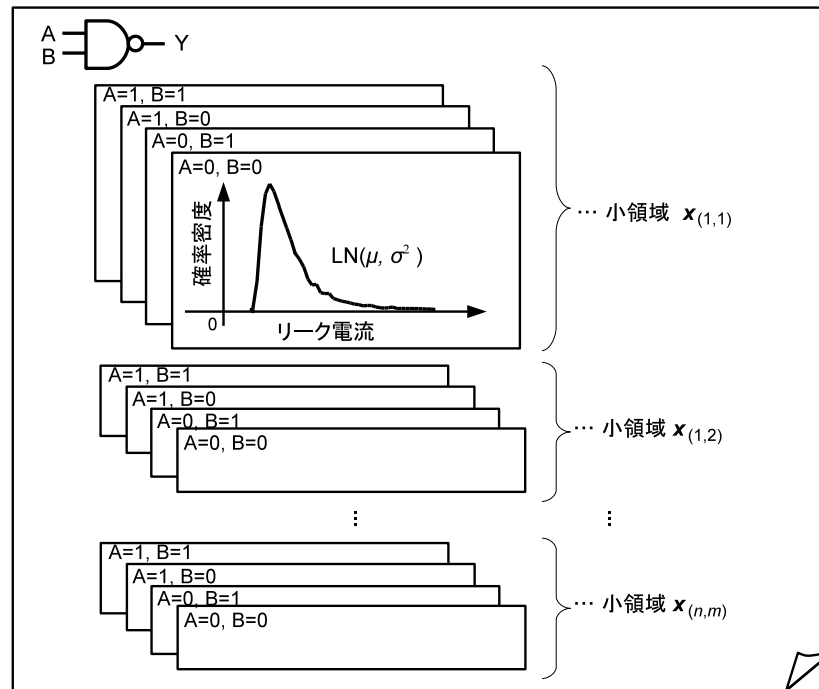


図 2.3: 2 入力 NAND ゲートのゲートレベル SLL の例．本ライブラリは，全パラメータ条件，全入力状態におけるリーク電流の確率密度分布を格納する．

数である．本節では，2 パラメータ数の問題，すなわち \mathbf{x} を ΔV_{thn} と ΔV_{thp} の 2 次元ベクトルとする場合を例として述べる．本推定手法は，チャネル長 L など，より多くの変数を推定する場合においても適用可能である．例えば，3 パラメータを推定する場合は，デバイスパラメータ空間を 3 次元として考える．

統計的リーク電流ライブラリ作成工程では，与えられた標準ロジックセルセットに対して，ゲートレベル SLL を生成する．ゲートレベル SLL は，情報として，各セルの入力状態毎の統計的リーク電流分布を，小領域毎に持つ．フリップフロップのように，リーク電流値が前時刻の内部状態に依存するセルに関しては，前時刻の内部状態を考慮する．2 入力 NAND ゲートを例にしたゲートレベル SLL の概念図を図 2.3 に示す．

ゲートレベル SLL を作成する処理の流れを，図 2.4 に示す．本工程では，リーク電流の統計的な分布を小領域 \mathbf{x} ，セル i ，入力状態 h の全ての組合せに対して求める．標準ロジックセルのリーク電流分布は，パラメータの大域ばらつき値を \mathbf{x} に固定し，ばらつきパラメータの局所ばらつきの分布に従う乱数値を加えたネットリストを作成して，Monte Carlo 回路シミュレーショ

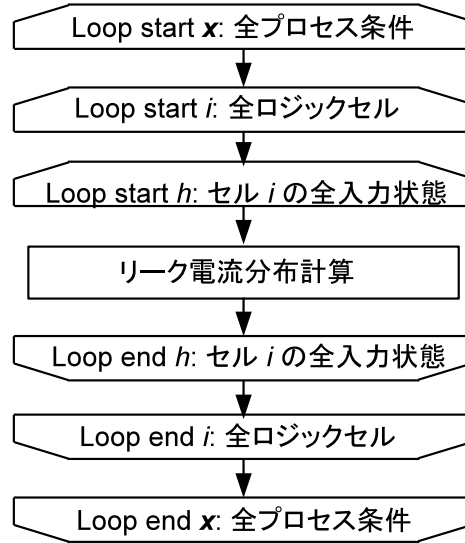


図 2.4: 統計的リーク電流ライブラリ生成工程の処理

ンにより計算する．Monte Carlo シミュレーションにおける試行回数は，要求されるパラメータ推定精度に依存する．試行回数については，本論文末の付録に示す結果を用いる．リーク電流分布は，計算したリーク電流値をライブラリに直接格納することも可能であるが，ゲートレベル SLL では，計算した標準ロジックセル毎のリーク電流分布を対数正規分布に近似して，ライブラリサイズを抑制する [30,32,51]．

本工程は，ロジックセル数と入力状態の組み合わせ数に応じて多くの計算時間が必要となるが，与えられた標準ロジックセルセットに対して一度行うのみでよい．

チップレベル統計的リーク電流分布の計算

続く工程で，IDDQ テストパターンに対するチップレベル SLL（対象回路の統計的リーク電流分布）を計算する．本工程の処理を，図 2.5 に示す．本工程は，全小領域 x ，テストパターン k の組合せに対して行う．対象回路の統計的リーク電流分布 I は全ゲートにおけるリーク電流値の総和として計算できる．

$$I = I_{\text{gate}_1} + I_{\text{gate}_2} + \cdots + I_{\text{gate}_N} \quad (2.1)$$

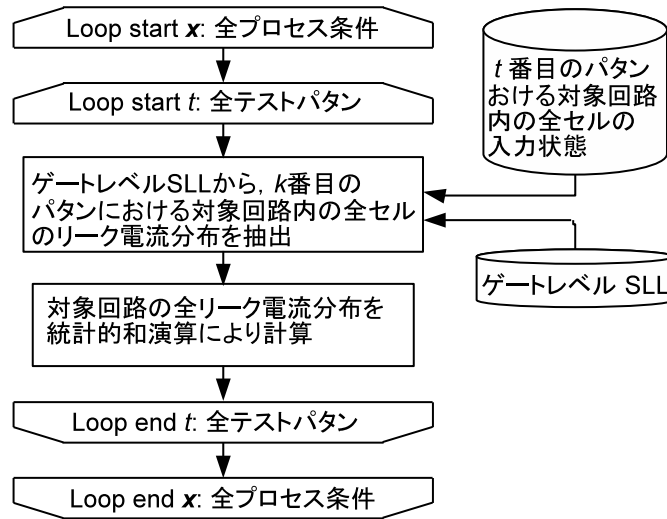


図 2.5: 統計的リーク電流分布計算工程の処理

ここで、 N は対象回路内のゲート数で、 I_{gate_j} は j 番目のゲートのリーク電流分布である。式 (2.1) の “+” は統計的 *SUM* 演算を表す。文献 [32,51] では、統計的リーク電流分布 I を解析的に解く手法が提案されているが、Monte Carlo シミュレーションを用いても I を得ることができる。今、各セルのリーク電流分布は、大域ばらつきを \mathbf{x} に固定して、局所ばらつき成分のみを考慮しているため、統計的 *SUM* 演算において、各パラメータ間の相関を考慮する必要がない。IDDQ 故障 ATPG により、各テストパターンにおける回路状態を得ており、ゲートレベル SLL からこれを参照することで対象回路の統計的リーク分布を計算できる。

ここで、小領域 \mathbf{x} における t 番目のテストパターンの IDDQ 測定時の対象回路のリーク電流を I として、対象回路のリーク電流分布の確率密度分布を $f_{(\mathbf{x};t)}(I)$ と定義する。

ベイズの定理に基づくデバイスパラメータの推定

本工程では、ベイズの定理を用いてチップ毎のデバイスパラメータを推定する。ベイズの定理を用いることで、IDDQ シグネチャから IDDQ 値を得る毎にデバイスパラメータの情報が更新され、最終的に、デバイスパラメータを高精度に推定できる。本推定手法では、推定対象チップが小領域 \mathbf{x} に属する確率を、小領域 \mathbf{x} の生起確率として計算する。生起確率が最大となる小

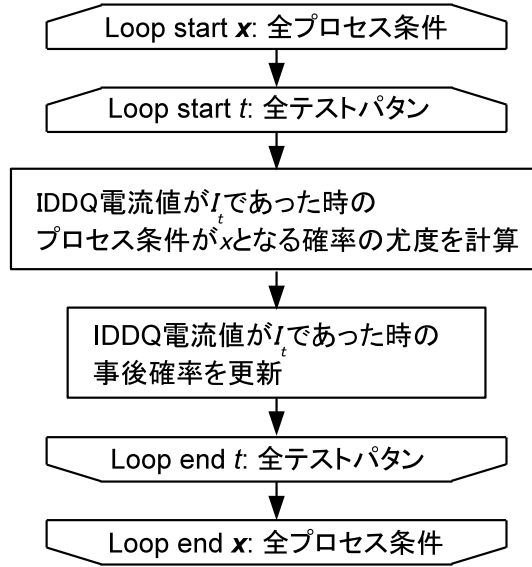


図 2.6: ベイズの定理に基づくデバイスパラメータ推定工程の処理

領域を，そのチップのデバイスパラメータ値とみなす．本工程は，各小領域 x ，各テストパタン t に対して行う．本工程の処理を図 2.6 に示す．

まず，推定対象チップが小領域 x に属する確率の事前確率を $P(x)$ とする．ベイズの定理から，最初の IDDQ テストパタン ($t = 1$) によって得られる IDDQ 電流値が I_1 であった時の事後確率 $P(x|I_1)$ は次のようになる．

$$P(x|I_1) = \frac{P(I_1|x)P(x)}{P(I_1)} \quad (2.2)$$

ここで， $P(x|I_1)$ は IDDQ 電流値 I_1 を観測したときにチップの属する小領域が x である確率を表す．また $P(I_1|x) \propto f_{(x,1)}(I_1)$ であり，式 (2.2) の分母は x によらず定数であるため，式 (2.2) は次のようになる．

$$P(x|I_1) \propto f_{(x,1)}(I_1)P(x) \quad (2.3)$$

$f_{(x,1)}$ は，チップレベル SLL から得られる．よって，確率 $P(x|I_1)$ は次のように計算できる．

$$P(x|I_1) = \frac{f_{(x,1)}(I_1)P(x)}{\sum_x f_{(x,1)}(I_1)P(x|I_1)} \quad (2.4)$$

ここで，式 (2.4) の分母は正規化定数である．また，式 (2.4) の分子の $f_{(x,1)}(I_1)$ は尤度を表す．

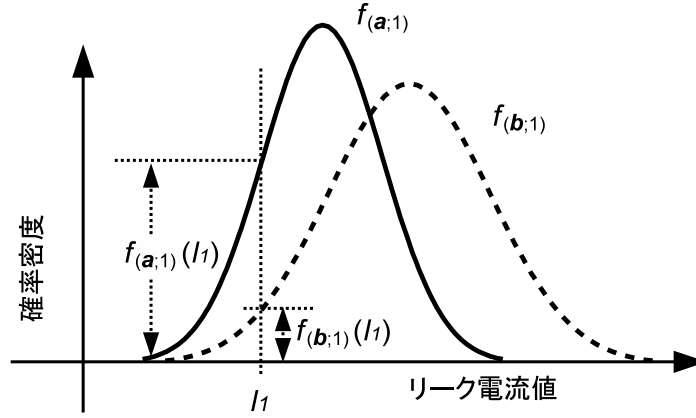


図 2.7: 尤度の概念図

図 2.7 に、尤度の概念を示す．図 2.7 において，IDDQ 電流値 I_1 が得られた時に，2 つの小領域候補 a ， b があつた場合を考える．尤度は，IDDQ 電流値 I_1 が得られた時に，そのチップがある小領域に属することがどれだけ尤もらしいかを表す．尤度は， $f_{(x;1)}(I_1)$ として計算される．図 2.7 において，小領域 a と b における尤度は，それぞれ $f_{(a;1)}(I_1)$ と $f_{(b;1)}(I_1)$ である．この例では，チップは小領域 a に属することがより尤もらしいといえる．

続いて，2 番目の IDDQ テストパターン ($t = 2$) によって得られる IDDQ 電流値 I_2 が観測された場合を考える．事前確率は $P(x|I_1)$ で，事後確率は $P(x|I_2)$ である．ベイズの定理から，式 (2.2) は次のようになる．

$$P(x|I_2) = \frac{P(I_2|x)P(x|I_1)}{P(I_2)} \quad (2.5)$$

式 (2.3) 同様，式 (2.5) は次のように表現できる．

$$P(x|I_2) \propto f_{(x;2)}(I_2)P(x|I_1) \quad (2.6)$$

式 (2.6) から，事後確率は次の式で計算できる．

$$P(x|I_2) = \frac{f_{(x;2)}(I_2)P(x|I_1)}{\sum_x P(x|I_2)f_{(x;2)}(I_1)} \quad (2.7)$$

式 (2.7) において， $P(x|I_2)$ は 2 つの IDDQ 電流値 I_1 と I_2 の情報を持つため，理想的には式 (2.4) の $P(x|I_1)$ よりも高精度である． $t \geq 3$ における事後確率 $P(x|I_t)$ も $t = 2$ と同様に計算できる．

$$P(x|I_k) = \begin{cases} \frac{f_{(x;1)}(I_1)P(x)}{\sum_x P(x|I_1)f_{(x;1)}(I_1)} & (t = 1) \\ \frac{f_{(x;t)}(I_t)P(x|I_{t-1})}{\sum_x P(x|I_t)f_{(x;t)}(I_t)} & (\text{otherwise}) \end{cases} \quad (2.8)$$

このように、ベイズの定理を用いて IDDQ 電流値の情報を更新することで、小領域毎の生起確率が更新される。よって、テストパターン数が多いほど、推定精度は高精度化する。

2.2.2 IDDQ シグネチャを用いたデバイスパラメータ推定実験

実験準備

本推定手法を用いることで、デバイスパラメータが推定できることを確認するため、ISCAS'89 ベンチマーク回路 [52] を用いたシミュレーション実験を行った。本節では、商用の 65-nm プロセスの標準セルライブラリを用いて設計した回路 s38584 に対する適用結果を示す。

IDDQ テストパターンは、商用 ATPG ツールを用いて生成する [53]。IDDQ テストのための故障モデルである疑似縮退故障モデルを対象としており、総テストパターン数は 49、故障検出率は 100 % である。

本実験では、次に示す手続きによって IDDQ シグネチャの測定を模擬した。

- (1) チップが属する小領域を決定する。
- (2) 対象回路の全ゲートに対して、ゲートレベル SLL 内のリーク電流分布からランダムにリーク電流値を取得する。リーク電流分布は、小領域とゲートの入力状態によって決まる。
- (3) 全ゲートにおけるリーク電流値の総和を計算することで、そのチップの全リーク電流値を得る。

上記の手続きをテストパターン数だけ繰り返し、テストパターン毎の IDDQ 電流値を得て、これを IDDQ シグネチャとする。

SPICE を用いた Monte Carlo 回路シミュレーションにより、セル毎の統計的リーク電流分布を計算する [54]。本シミュレーションは、室温を想定して 1,000 回行った。標準セルライブラリセットには 57 セルあり、これら全セルの入力組合せの和は 1,218 である。本実験では、nMOS トランジスタと pMOS トランジスタのしきい値電圧がばらつくと仮定した。両者の分布は独立であり、局所ばらつき成分は、Pelgrom モデルに従うとした [55]。本節では、nMOS および pMOS トランジスタの大域ばらつき成分を ΔV_{thn} および ΔV_{thp} と表記する。 ΔV_{thn} と ΔV_{thp} の範囲は、それぞれ -80 mV から $+80 \text{ mV}$ とする。小領域 x は ΔV_{thn} と ΔV_{thp} の 2 次元ベクトルである。SPICE シミュレーションでは、ゼロバイアス時のしきい値電圧変動量 DELVT0 を用いて ΔV_{thn} と ΔV_{thp} を考慮した。デバイスパラメータ空間を、それぞれ

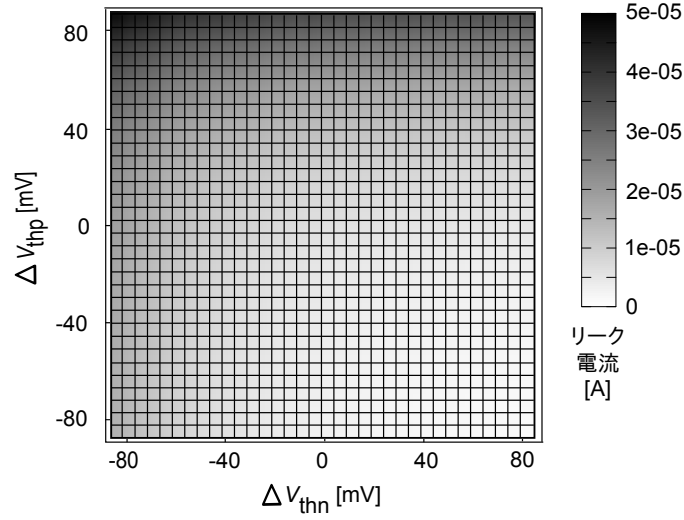


図 2.8: 小領域毎のテストパターン $t = 1$ の時の s38584 のリーク電流マップ

5mV 単位で 1089 個の小領域に分割する．2.2.1 節で述べたように，5mV が本実験における推定の最小分解能であるため，推定が正しく行われた場合の推定精度は 5mV である．デバイスパラメータ空間をより小さく分割することで，より高精度な推定結果が得られる．

以下では，小領域 x を $(\Delta V_{thn}, \Delta V_{thp})$ のように括弧を用いて表記する．例えば， $\Delta V_{thn} = 10\text{mV}$ ， $\Delta V_{thp} = -10\text{mV}$ の場合は， $(10\text{mV}, -10\text{mV})$ と表記する．

ベイズの定理を用いたパラメータ推定工程における $P(x)$ の初期分布は， -80mV から 80mV の範囲における一様分布と仮定した．

図 2.8 に，s38584 のテストパターン $t = 1$ の時のリーク電流マップを示す．図 2.9 に，同回路の統計的リーク電流分布の対数正規分布の μ と σ を示す．図 2.9 はチップレベル SLL から得ることができる．これらの図で，縦軸と横軸はそれぞれ ΔV_{thn} と ΔV_{thp} である．図の右下コーナーは nMOS と pMOS トランジスタのスイッチング速度が遅い SS コーナーであり，これに対して左上コーナーは，両トランジスタのスイッチング速度が速い FF コーナーを表す．同様に，左下と右上のコーナーはそれぞれ FS，SF コーナーである．

図 2.8 と図 2.9(a) では， -45 度の線上の SS コーナーから FF コーナーに向かって，リーク電流値が指数関数的に増加している．特に，上辺と左辺の両辺においてリーク電流値が大きい．上辺においては pMOS トランジスタのスイッチング速度が速く，左辺においては nMOS トランジスタのスイッチング速度が速い傾向にあるためである．スイッチング速度が速いトランジ

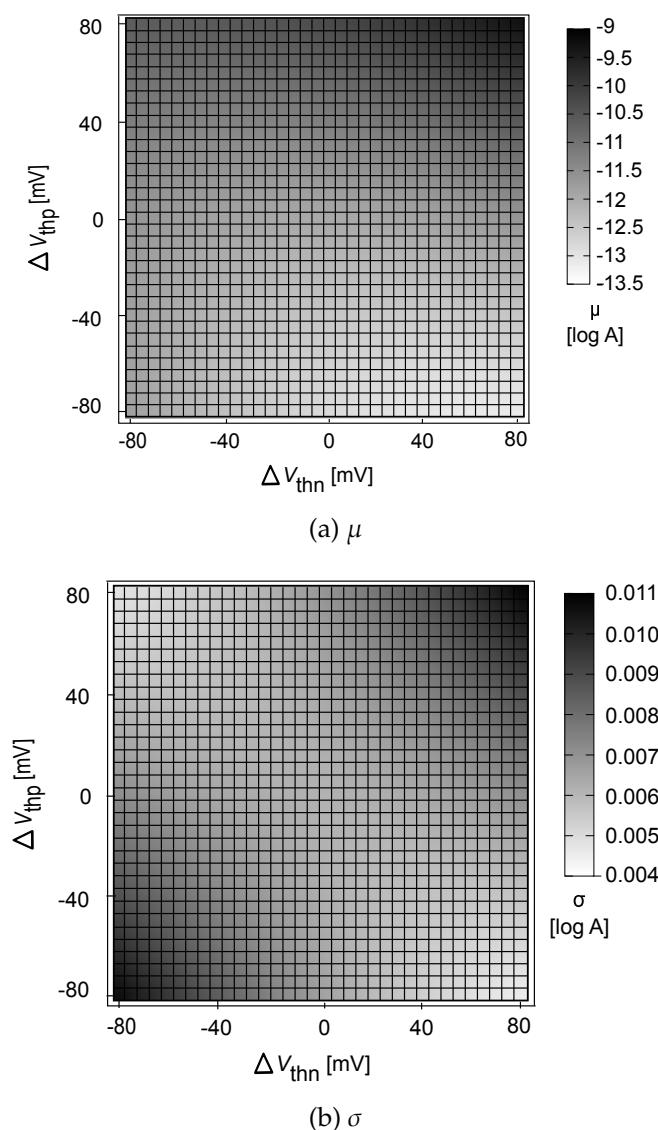
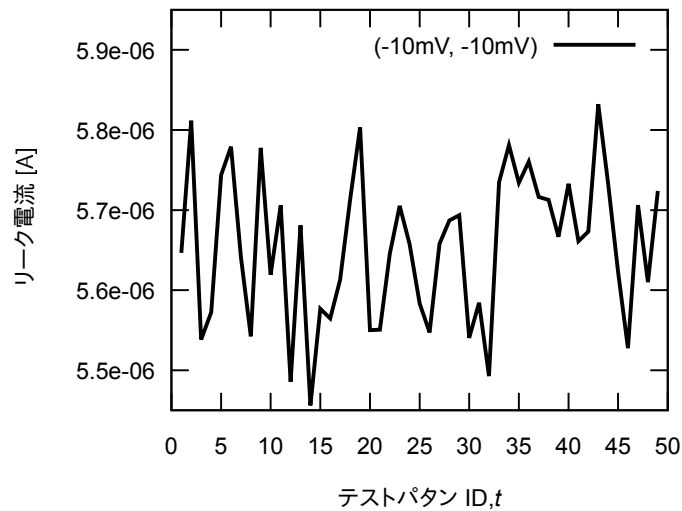


図 2.9: 各小領域における，テストパターン $t = 1$ の時の全リーク電流の対数正規分布の μ と σ

スタは，リーク電流が大きい傾向にある．また，図 2.9(b) では SF, FS コーナーではリーク電流値の分散が大きくなり，異なる傾向が見られる．これは，SF, FS コーナーでは，nMOS と pMOS トランジスタのリーク電流値が異なり，わずかな局所ばらつき成分の増減により，リーク電流値が大きく変動するためである．

表 2.1: 各工程における CPU 時間

工程	処理時間 (分)
ゲートレベル SLL 生成	17,640
チップレベル SLL 計算	2,880
ベイズの定理に基づくデバイスパラメータ推定	1

図 2.10: $(-10\text{ mV}, -10\text{ mV})$ における IDDQ シグネチャ

デバイスパラメータ推定結果

提案手法を C 言語と Ruby で実装し, IDDQ シグネチャからデバイスパラメータを推定できることを確認する実験を行った. 計算機実験は, Intel(R) Xeon(R) プロセッサ (CPU: 2.93 GHz, 8 MB Cache) 上で実行した. 提案手法における各工程の CPU 時間を表 2.1 にまとめる. 表 2.1 はシングルスレッドによる結果を記載しているが, ゲートレベル SLL, チップレベル SLL の計算時間は, それぞれ 294 時間, 48 時間かかるが, 並列処理することで CPU 処理時間の短縮が可能である. また, ゲートレベル SLL 生成は, 与えられた標準セルセット毎に 1 回実施するのみである. また, チップレベル SLL 計算は, ゲートレベル SLL を基に推定対象回路毎に 1 回行うのみである.

提案するデバイスパラメータ推定手法を, 正解の小領域が $(-10\text{ mV}, -10\text{ mV})$ の場合の推定に適用する. 図 2.10 に $(-10\text{ mV}, -10\text{ mV})$ の時の模擬チップの IDDQ シグネチャを示す. 横軸がテストパターン ID で, 縦軸がリーク電流値である.

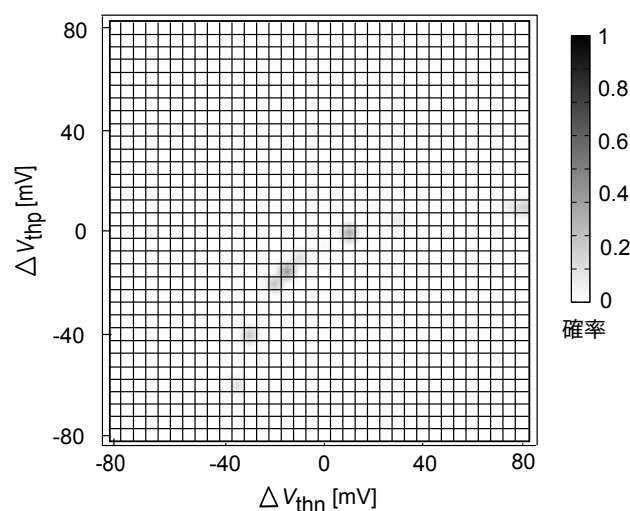


図 2.11: $(-10 \text{ mV}, -10 \text{ mV})$ に対する，最後のテストパタンの時のデバイスパラメータ推定結果

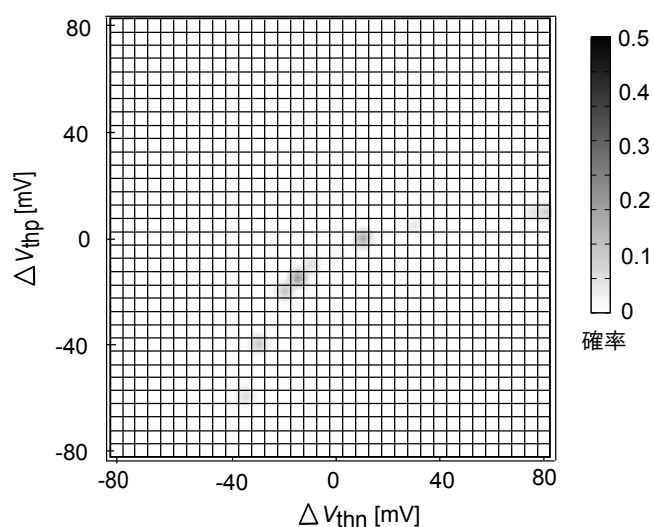


図 2.12: テストパターン $t = 3$ の時の $(-10 \text{ mV}, -10 \text{ mV})$ に対するデバイスパラメータ推定結果

図 2.11 に，全ての IDDQ 電流を推定に用いた後の推定確率 $P(\mathbf{x}|\mathbf{I}_{49})$ を示す．この図の横軸と縦軸は，それぞれ ΔV_{thn} と ΔV_{thp} である．図中の四角は， $10 \text{ mV} \times 10 \text{ mV}$ の小領域を表す．白色の小領域は推定確率が 0 であることを表し，推定確率が高くなるほど，白色から黒色となるよう色付けを行っている．図 2.11 から，正解領域 $(-10 \text{ mV}, -10 \text{ mV})$ が推定確率 100 % となっている．

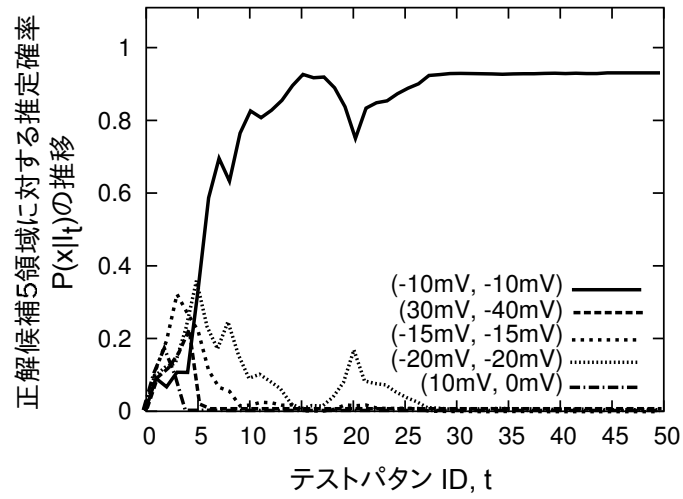


図 2.13: 正解候補 5 領域における，テストパターン t に対する推定確率 $P(\mathbf{x}|I_t)$ の推移

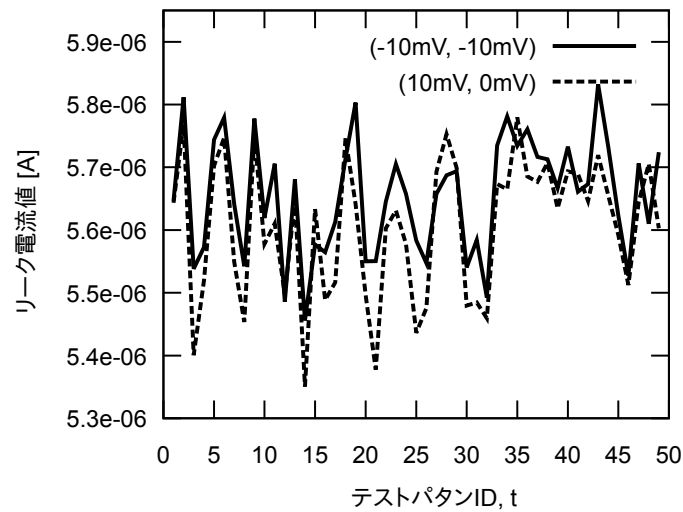


図 2.14: (10 mV, 10 mV) と (-10 mV, -10 mV) の IDDQ シグネチャ

おり，正しく推定できていることが分かる．図 2.12 に，テストパターン $t = 3$ の時の推定確率 $P(\mathbf{x}|I_3)$ を示す．この時点では，正解領域 (-10 mV, -10 mV) を含む 10 個の小領域の候補があり，(-15 mV, -15 mV) が最も高い推定確率となっている．

図 2.13 に，テストパターン t における推定確率 $P(\mathbf{x}|I_t)$ の推移を示す．横軸はテストパターン ID，縦軸は推定確率である．実線は正解領域 (-10 mV, -10 mV)

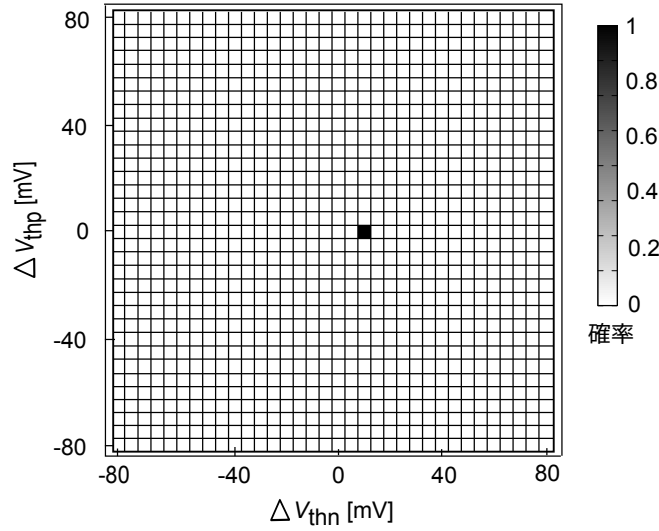


図 2.15: (10 mV, 0 mV) における，最終テストパタンの時のデバイスパラメータ推定結果

に対する推定確率の推移である．破線は，図 2.12 の正解領域以外の推定候補 9 領域の上位 4 領域に対する推定確率の推移である．実線において，確率 $0.0009 (= 1/(33 \times 33))$ から開始し 30 パタン以内で 1 まで上昇している．1 に収束するまで，推定確率は単調増加ではなく増減している．これは，局所ばらつき成分の影響と，同程度のリーク電流値となる小領域が複数存在するためである．しかし，テストパタン数を増やすことで，推定候補となる小領域を絞り込むことができ，最終的に正解領域に収束している．

続いて，(10 mV, 0 mV) に属する模擬チップに提案手法を適用した結果を示す．図 2.14 に，(10 mV, 0 mV) と (-10 mV, -10 mV) の時の模擬チップの IDDQ 電流シグネチャを示す．実線と破線は，それぞれ (-10 mV, -10 mV) および (10 mV, 0 mV) にそれぞれ対応する．図 2.14 の実線は図 2.10 と同様である．図 2.14 から，IDDQ シグネチャは大部分が重なっており，これら 2 つの IDDQ シグネチャからデバイスパラメータを推定することは容易ではない．

図 2.15 に，(10 mV, 0 mV) に対する，最終テストパタンの推定確率 $P(x|I_{49})$ を示す．正解領域である (10 mV, 0 mV) が正しく推定されている．図 2.15 と図 2.11 から，提案手法を用いることで，重なりが多い IDDQ 電流シグネチャであったとしても，デバイスパラメータの異なる 2 チップを区別できることが分かる．また，図 2.12 を確認すると，(10 mV, 0 mV) もまた，正解領域の候補となっている．

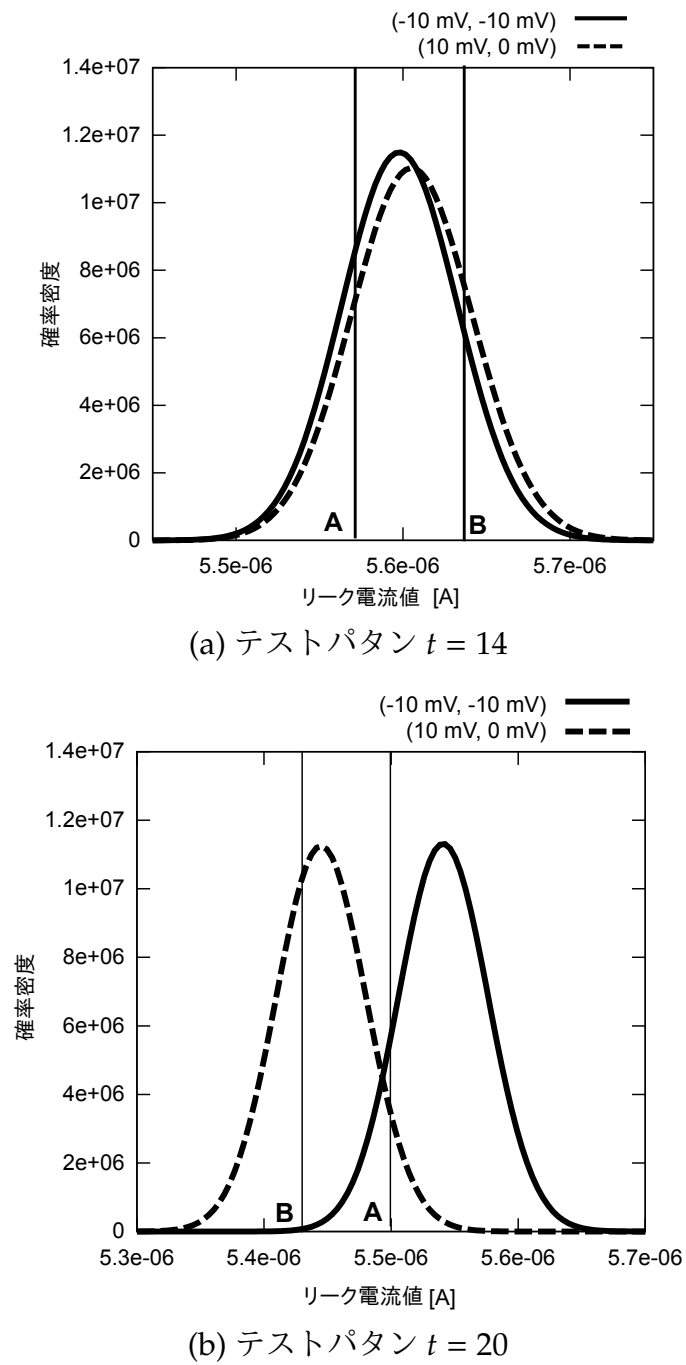


図 2.16: $(-10 \text{ mV}, -10 \text{ mV})$ と $(10 \text{ mV}, 0 \text{ mV})$ におけるリーク電流の確率密度分布

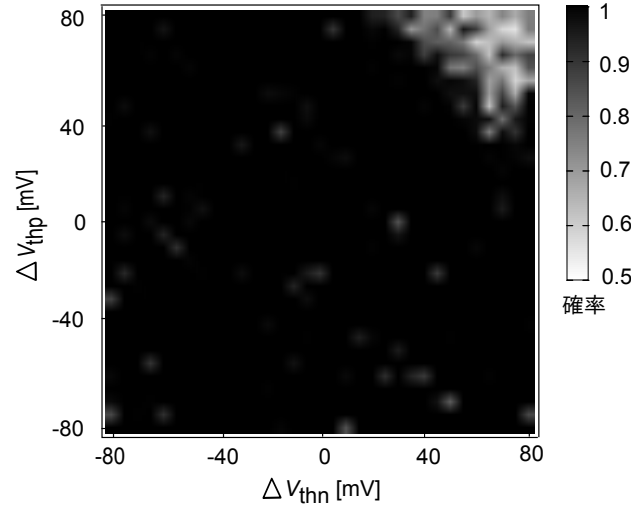


図 2.17: 最終テストパターンにおけるの全小領域のデバイスパラメータ推定結果

図 2.16 に, $t = 14$ と $t = 20$ の時の $(-10 \text{ mV}, -10 \text{ mV})$ および $(0 \text{ mV}, -10 \text{ mV})$ のリーク電流の確率密度分布を示す. 横軸はリーク電流値, 縦軸は確率密度である. これらの分布は, ゲートレベル SLL から得ることができる. 実線と破線は, $(-10 \text{ mV}, -10 \text{ mV})$, $(0 \text{ mV}, -10 \text{ mV})$ に対応する. 図中横軸の A と B は, $(-10 \text{ mV}, -10 \text{ mV})$ および $(0 \text{ mV}, -10 \text{ mV})$ における, それぞれのテストパターンで得られた IDDQ 電流値である. 図 2.14 において, $t = 14$ の時の両領域における IDDQ 電流値は非常に近く, 図 2.16(a) においても, 確率密度分布の大部分が重なっている. 一方, 図 2.14 において, $t = 20$ の時の両領域における IDDQ 電流値は離れており, 図 2.16(b) のように重なりが少ない. このように, IDDQ シグネチャの IDDQ 電流値の大部分が重なりあったとしても, 重なり合わない部分は確率密度が離れているため, バイズの定理の尤度計算から二つの模擬チップを区別できる.

最後に, 提案手法を 1,089 個の模擬チップに対して適用した結果を示す. これらの模擬チップのデバイスパラメータは, それぞれの小領域 (全 1,089 領域) に対応している. この実験で, 任意の小領域においても, 提案手法が正しく推定できることを示す.

図 2.17 に, 全小領域 x に対する最終の推定結果 $P(x|I_{49})$ を示す. 図の各小領域は, その領域に対する推定確率を示している. 推定結果が 100 % ならば, 領域の色は黒になる. 図 2.17 から, ほとんどの領域において推定確率が 100 % となっており, 提案手法が正しく推定できていることがわかる. し

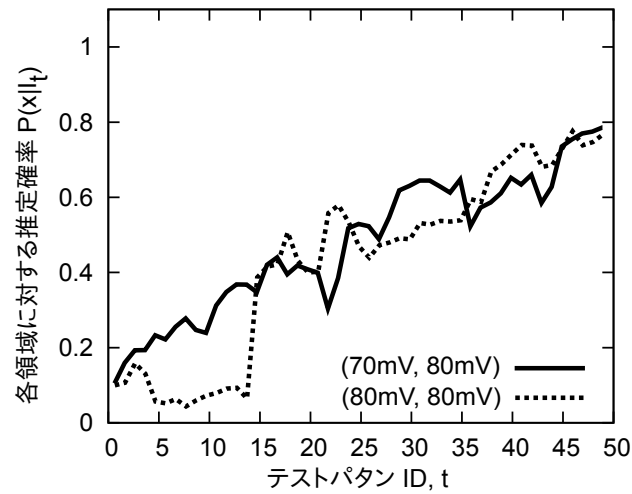


図 2.18: (70 mV, 80 mV) と (80 mV, 80 mV) における, テストパターン t に対する推定確率 $P(x|I_t)$ の推移

かし, 図の右上コーナーに, 推定確率が 100% となっていない領域が存在する. 図 2.18 に (70 mV, 80 mV) と (80 mV, 80 mV) に対する推定確率の推移を示す. 図 2.13 とは異なり, 推定確率が 100% に収束していない. この理由は次の 2 点である. (1) 全てのテストパターンにおいて図 2.16(a) のように確率密度分布の大部分が重なっている. (2) 図 2.9(b) で示すように, リーク電流分布の標準偏差が大きいことによる. 本実験において, $t = 49$ の時の推定確率は 50% を上回っており, 正解領域が最大の推定確率となっている. 以上から, 提案手法は任意の小領域において正しく推定できることが分かる.

2.2.3 まとめ

本節では, IDDQ シグネチャを用いたデバイスパラメータ推定手法を提案した. 本推定手法は, ベイズの定理を用いて IDDQ シグネチャからチップ毎のデバイスパラメータ値を導出する. 提案手法は, デバイスパラメータ空間を小領域に分割し, 小領域毎の生起確率を計算する. 最も高い生起確率である小領域をそのチップのデバイスパラメータ値とする. 本推定手法は, 推定のための情報を全て IDDQ テストフローから得るため, 既存の特性ばらつき推定手法と異なり, 推定のための特別な回路の追加や測定は不要である. 計算機実験では, デバイスパラメータ空間を 5 mV 刻みで分割し, 提案手法を用いることで, トランジスタのしきい値ばらつきの全域ばらつき成分を,

推定の最小分解能である 5 mV 精度で推定できることを示した.

2.3 最大動作周波数テストの枠組みを用いた手法

本節では、最大動作周波数 (F_{\max}) テストの枠組みを用いたデバイスパラメータ推定手法を提案する。2.2 節で提案した推定手法同様、本推定手法も、第 3, 4 章で述べる適応型テストに向けたデバイスパラメータ推定手法である。IDDQ テストと F_{\max} テストは独立したテストであるため、両手法を同時に適用することで、デバイスパラメータ推定結果の検証が可能である。ここでは、離散ベイズ推定法 (Discrete Bayesian Estimation) を用いた手法 (DBE 方式推定) と最尤推定法 (Maximum Likelihood Estimation) を用いた手法 (MLE 方式推定) の 2 手法を提案する。これらの推定手法は、 F_{\max} テストにおける測定で得られる情報と F_{\max} テストで活性化されるパスの統計的遅延分布をライブラリとして用いる。2.2 節で提案した手法同様、追加回路の搭載が不要である。本節では、 F_{\max} テストの枠組みを用いた手法を用いることで、デバイスパラメータを高精度に推定できることを示す。

2.3.1 最大動作周波数テスト

本推定手法は F_{\max} テストの枠組みを用いる。本節では、 F_{\max} テストについて概説する。 F_{\max} テストは、 F_{\max} 値を特徴値として用いる [45]。チップ製造後、遷移遅延故障テストパターン [23] を用いて F_{\max} テストをチップ毎に行う。パス遅延故障モデルがパスに故障があることを仮定しているのに対し、遷移遅延故障モデルは、論理ゲートの入出力を構成する一つの信号線が遅延することを仮定した故障モデルである。遷移遅延故障テストパターンは、遷移遅延故障モデルを対象に生成されたテストパターンである。 F_{\max} テストでは、チップ毎に 1 つの F_{\max} 値が得られ、図 2.19 のように他の大多数のチップと異なる異常な F_{\max} 値が得られた場合、そのチップを不良と判定する異常値選別手法である。IDDQ テストの結果と組み合わせる事で、多変量解析によるテストへも応用できる [34]。その他、最小動作電圧 (MinVDD) を用いる異常値選別手法も提案されている [34]。

2.3.2 最大動作周波数テストの枠組みを用いたデバイスパラメータ推定

本節では、 F_{\max} テストの枠組みを用いたデバイスパラメータ推定手法を提案する。推定の手法として、DBE 方式推定と MLE 方式推定を提案する。

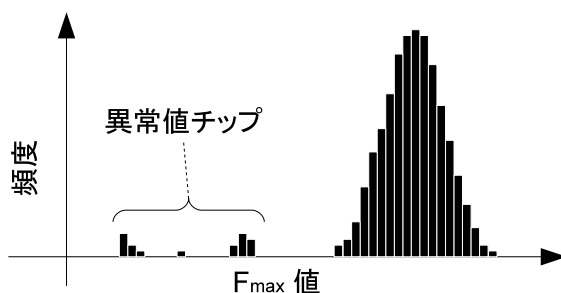


図 2.19: F_{\max} テストにおける異常値選別手法

本推定手法は、ベイズ推定と最尤推定法を用いることで高精度にデバイスパラメータを推定できる点が特徴である。

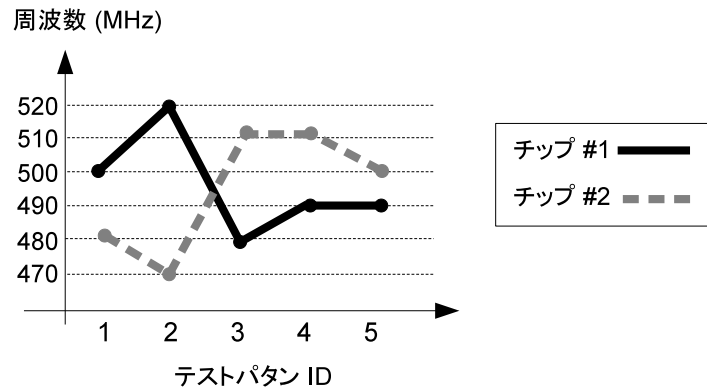
推定手法の概要

DBE 方式推定では、想定するデバイスパラメータ空間を小領域に分割する。テストパターン毎に活性化されるパスのパス遅延値を各小領域に計算し、これを統計的最大パス遅延ライブラリ (SMPDL) としてコンパクトに保存する。 F_{\max} テストでは、テストパターン毎の F_{\max} 値を測定し、測定した最大周波数値と SMPDL からベイズの定理を用いて、小領域毎のデバイスパラメータ値の生起確率を推定する。

MLE 方式推定では、デバイスパラメータが正規分布に従うと仮定し、最尤法を用いて観測される最大周波数値の生起確率を最大化するように、デバイスパラメータの平均値と標準偏差を計算する。

MLE 方式推定は、デバイスパラメータが正規分布に従うと仮定して推定する。これに対し、DBE 方式推定はデバイスパラメータの分布について仮定を行わないため、正規分布等に従わない分布のデバイスパラメータでも高精度に推定できる。一方、MLE 方式推定は正規分布仮定により高速な推定が可能となる。ただし、デバイスパラメータが正規分布に従わない場合は、正規分布の近似誤差により推定精度が悪化する可能性がある。よって、推定計算時間と要求される推定精度、デバイスパラメータの形状によって、DBE 方式推定と MLE 方式推定から、より良い推定手法を選択することができる。

両手法とも、テストパターン毎の F_{\max} 値を必要とする。これらの数値を得るために、本推定手法では F_{\max} シグネチャを新たに導入する。 F_{\max} シグネチャは、テストパターン毎の F_{\max} 値のベクトルとして定義する。既存の F_{\max}

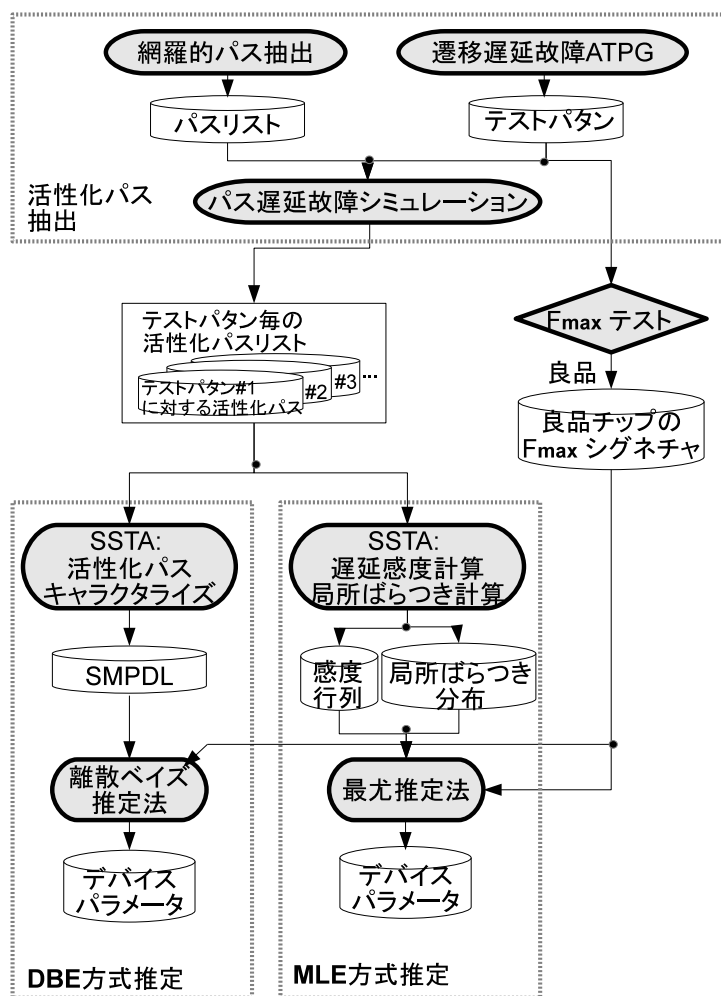
図 2.20: 2 チップにおける F_{\max} シグネチャの例

テストでは、チップ毎に最大動作周波数値 F_{\max} を 1 つだけ得るのに対し、本推定手法における F_{\max} テストではテストパターン毎に F_{\max} 値を得る。

図 2.20 に F_{\max} シグネチャの例を示す。ここでは、2 チップに対する F_{\max} シグネチャを考える。本例の F_{\max} シグネチャは、5 テストパターン分の F_{\max} 値で構成されている。既存の F_{\max} テストでは、チップ #1 とチップ #2 に対して F_{\max} 値 470 MHz, 480 MHz を取得するのみである。しかし、本推定手法における F_{\max} テストでは、テストパターン毎の F_{\max} シグネチャを要素毎のベクトルとして取得する。すなわち、チップ #1 に対しては (500 MHz, 520 MHz, 480 MHz, 490 MHz, 490 MHz) を取得し、チップ #2 に対しては (480 MHz, 470 MHz, 510 MHz, 510 MHz, 500 MHz) を得る。 F_{\max} シグネチャを得るために、ATE 上でテストパターン毎の線形探索を行うことを想定している。

先端プロセスでは、デバイスパラメータが大きくばらつくため、図 2.20 に示すように、2 つの F_{\max} シグネチャの傾向はテストパターン毎に異なる。本推定手法は、この特徴を利用してデバイスパラメータを推定する。2 パターン目において nMOS トランジスタが多く遷移するように活性化されており、3 パターン目において、pMOS トランジスタが多く遷移するように活性化されている場合を考える。この条件で図 2.20 の F_{\max} シグネチャが得られたとすれば、チップ #1 は FS 条件 (Fast-Slow: nMOS と pMOS がそれぞれ Fast 条件, Slow 条件にある) ことを示唆する。一方で、チップ #2 は SF 条件 (Slow-Fast) にあることを示唆する。本推定手法では、複数テストパターンでの最大周波数とベイズ推定、最尤推定法を組み合わせることで、チップ毎のデバイスパラメータを高精度に推定する。

図 2.21 にこれらの推定手法の全体フローを示す。本推定手法では、 F_{\max}

図 2.21: F_{\max} テストの枠組みを用いたデバイスパラメータ手法の全体フロー

テストで活性化されるパスをテストパターン毎に抽出し、統計的パス遅延分布、またはパス遅延感度及び局所ばらつき分布の計算を行う。 F_{\max} テストでは、 F_{\max} シグネチャとしてテストパターン毎の F_{\max} 値を得る。これらの情報に対して、離散ベイズ推定法、最尤推定法を用いて、チップ毎のデバイスパラメータを推定する。 F_{\max} テストでは、NNR (Nearly Neighbor Residual) 手法 [56] 等の統計処理によって、テスト対象チップの良品/不良品の判定を行うことができる。推定対象チップに故障がある場合、デバイスパラメータが正しく得られない可能性がある。本節では、良品チップの F_{\max} シグネチャが用いられると仮定し、本推定手法の原理を説明する。

活性化パス抽出

提案手法では、まず、 F_{\max} テストにおいて活性化されるパスのリストをテストパターン毎に生成する。この工程は、両手法共通である。一般に、テスト対象回路のテスト網羅性を上げるために、 F_{\max} テストは遷移遅延故障テストパターンを用いて行われる [34]。遷移遅延故障テストパターンは自動テストパターン生成 (ATPG) ツールを用いて生成される。同時に、統計的静的タイミング解析 (SSTA) ツールを用いて、テスト対象回路のパスを網羅的に抽出する。続いて、遷移遅延故障パスと抽出されたパスを用いて、パス遅延故障シミュレーションを行う。提案手法では、 F_{\max} テストにおいて、始点フリップフロップから終点フリップフロップまで確実に信号伝搬しているパスのリストが必要である。よって、パス遅延故障シミュレーションでは、検出条件をロバスト検出条件とする [57]。ロバスト検出条件は、テスト対象パスのパス外入力が常に非制御値になっている検出条件である。つまり、2 入力 AND ゲートを考えた場合、パス外入力が非制御値である論理値 1 であれば、テスト対象パス上の入力信号遷移は AND ゲートの出力へ伝搬する。テストパターン毎のロバスト検出可能なパスのリストを生成し、各推定手法の入力とする。ロバスト検出条件で検出と判定されたパスは、活性化パスリストとしてまとめられ、DBE 方式推定、MLE 方式推定の入力情報となる。

本工程の処理は、市販 EDA ツールを組み合わせることで実現できる。

離散ベイズ推定法を用いたデバイスパラメータ推定

DBE 方式推定では、想定するデバイスパラメータの変動域を小領域に分割する。例えば、推定対象となるデバイスパラメータが 2 つの場合、その変動領域は 2 次元となる。以下、本節では x は小領域空間を表し、 n 次元ベクトルである。 n は推定するデバイスパラメータの数である。DBE 方式推定では、先の工程で得た活性化パスリストを入力として、小領域毎、テストパターン毎の活性化パスの遅延分布を計算する。これを尤度関数として、 F_{\max} シグネチャを得る確率を小領域毎に離散ベイズ推定法により計算する。DBE 方式推定では、小領域毎の生起確率としてデバイスパラメータを推定する。

まず、小領域毎、テストパターン毎の活性化最大パス遅延ライブラリ (SMPDL) を得る。SMPDL を求める手続きを図 2.22 に示す。SMPDL は、情報として、テストパターン毎に活性化されたパスリストの最大パス遅延分布を小領域毎に持つ。SSTA ツールを用いて、活性化パス毎のパス遅延分布を計算し、統計的 MAX 演算によって、テストパターン毎の最大パス遅延分布を計算

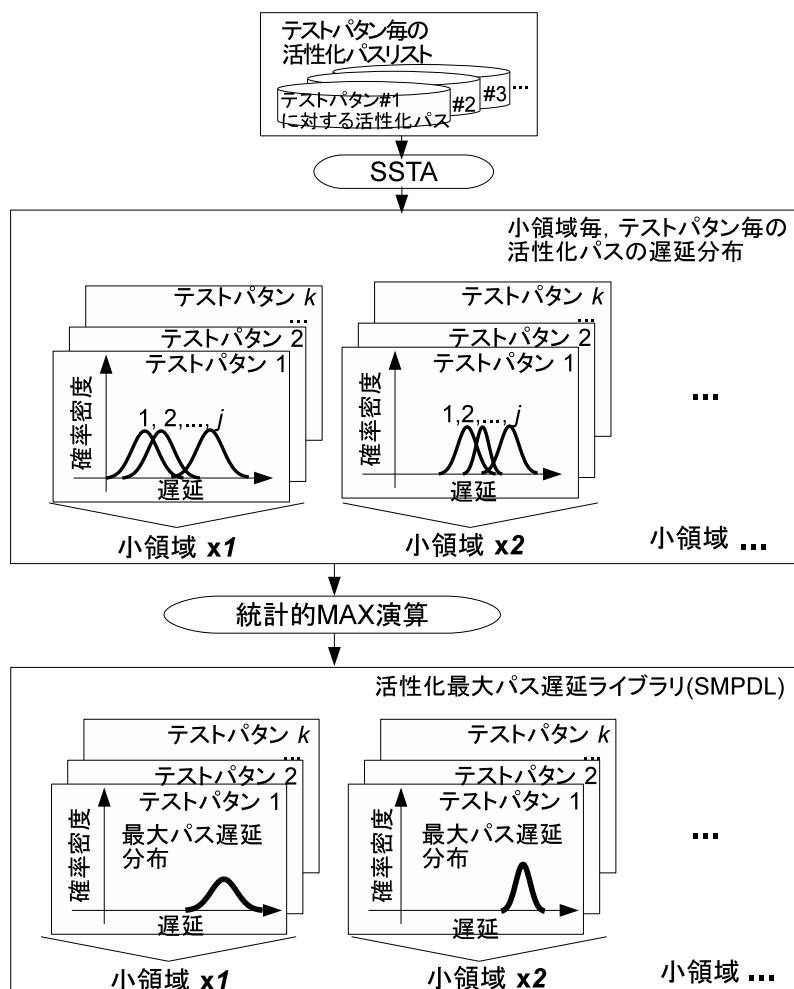


図 2.22: SMPDL 生成の手続き

する。これらの計算は、小領域毎に行われる。デバイスパラメータ値は各小領域に対応しているため、パス遅延分布は局所ばらつきのみによる分布である。

SMPDL を求めるために、小領域 x において、 k 番目のテストパターンによって活性化されるパスの最大パス遅延分布 $d_{\max(x,k)}$ を計算する必要がある。まず、小領域 x における、 k パターン目で活性化される j 番目のパスの遅延分布を $d_{(x,j,k)}$ とする。ここで、テストパターン数は l であり、 $1 \leq k \leq l$ である。文

献 [18] より, パス遅延分布 $d_{(x,j,k)}$ は一次の標準形で表すことができる.

$$d_{(x,j,k)} = d_{ave(x,j,k)} + \sum_{i=1}^n s_{(j,p_i)} \Delta p_i + N(0, \sigma_{rnd(j,k)}^2) \quad (2.9)$$

ここで, $d_{ave(x,j,k)}$ はパス遅延 $d_{(x,j,k)}$ の平均値である. Δp_i は, i 番目のデバイスパラメータを表す変数であり, \mathbf{x} の i 番目の要素である. ここで, $1 \leq i \leq n$ である. $s_{(j,p_i)}$ は, Δp_i に対するパス遅延感度である. $N(0, \sigma_{rnd(j,k)}^2)$ は局所ばらつきによるパス遅延の変動を表し, 平均値が0, 標準偏差が $\sigma_{rnd(j,k)}^2$ の正規分布である. $N(0, \sigma_{rnd(j,k)}^2)$ は局所ばらつきであるため, \mathbf{x} に依存しない.

小領域 \mathbf{x} における k パタン目の活性化パスの最大パス遅延分布 $d_{max(x,k)}$ は, 統計的 MAX 演算を用いて次のように表せる.

$$d_{max(x,k)} = \max_j(d_{(x,j,k)}) \quad (2.10)$$

式 (2.10) より, チップが小領域 \mathbf{x} に属する時, k パタン目によって活性化されるパス群の最大パス遅延 d_{max} は, 確率密度関数から $f_{(x,k)}(d_{max})$ と求められる. SMPDL は, 全小領域における, 全テストパタンに対する最大パス遅延分布を持つ. 統計的 MAX 演算によって得られる最大パス遅延分布は正規分布に従うと仮定する.

次に, SMPDL と F_{max} シグネチャを入力として, 離散ベイズ推定法を用いて, 各チップのデバイスパラメータ値を推定する. ベイズの定理を用いることで, F_{max} シグネチャから F_{max} 値を得る度に小領域 \mathbf{x} の生起確率が更新され, 最終的に高精度にデバイスパラメータ値を得ることができる. 本工程は, 各小領域 \mathbf{x} , 各テストパタン k に対して行われる. 本工程では, ベイズの定理を用いて, あるチップにおける小領域 \mathbf{x} の生起確率を得る. 小領域 \mathbf{x} の生起確率を事前確率として, F_{max} 値から, あるチップの小領域 \mathbf{x} となる尤度を計算する. テストパタン k が増加することで, F_{max} 値 d_k が得られた後の事後分布が更新される.

ベイズの定理から, k 番目のテストパタンで得られる F_{max} 値 d_k が得られた時の事後確率 $P(\mathbf{x}|d_k)$ は次のようになる.

$$P(\mathbf{x}|d_k) = \frac{P(d_k|\mathbf{x})P(\mathbf{x}|d_{k-1})}{P(d_k)} \quad (2.11)$$

分母は, 正規化定数であり, $P(d_k|\mathbf{x})$ は, 小領域 \mathbf{x} の SMPDL と事前分布 $P(\mathbf{x}|d_{k-1})$ から計算される尤度である. $k = 1$ の時の初期事前分布 $P(\mathbf{x}|d_0)$ は $P(\mathbf{x})$ とす

る．ここで， $P(\mathbf{x}|d_k) \propto f_{(\mathbf{x},k)}(d_k)$ であり，式 (2.11) の分母はいかなる \mathbf{x} に対しても定数であるため，式 (2.11) は次のように変形できる．

$$P(\mathbf{x}|f_k) \propto f_{(\mathbf{x},k)}(d_k)P(\mathbf{x}|d_{k-1}) \quad (2.12)$$

$f_{(\mathbf{x},k)}$ は式 (2.10) から得られる．よって，確率 $P(\mathbf{x}|d_k)$ は次のように計算できる．

$$P(\mathbf{x}|d_k) = \frac{f_{(\mathbf{x},k)}(d_k)P(\mathbf{x}|d_{k-1})}{\sum_{\mathbf{x}} f_{(\mathbf{x},k)}(d_k)P(\mathbf{x}|d_k)} \quad (2.13)$$

ここで，式 (2.13) の分母は正規化定数である．理想的には， F_{\max} シグネチャの F_{\max} 値 d_k が更新される毎に，事後分布はより高精度化される．よって，各領域の生起確率は，そのチップが属する特定の小領域のみが高くなり，それ以外は0%に収束することが期待される．

最尤推定法を用いたデバイスパラメータ推定

MLE 方式推定では， F_{\max} シグネチャが得られる確率をデバイスパラメータの平均と標準偏差の関数とみなして，これを尤度関数とする．最尤推定法を用いて，尤度関数を最大化する平均と標準偏差を求める．

まず，デバイスパラメータの平均と標準偏差の関数として， F_{\max} シグネチャが得られる確率を導出する．DBE 方式推定が小領域毎に生起確率を推定していたのに対し，MLE 方式は小領域を考慮しない．ある小領域 \mathbf{x} のみに限定しない場合，式 (2.9) は次のようになる．

$$d_{(j,k)} = d_{ave_{(j,k)}} + \sum_{i=1}^n s_{(j,p_i)} \Delta p_i + N(0, \sigma_{rnd_{(j,k)}}^2) \quad (2.14)$$

続いて，式 (2.10) 同様， k 番目のテストパターンで活性化されるパスの最大パス遅延分布 d_k は次のようになる．

$$\begin{aligned} d_k &= \max_j (d_{(j,k)}) \\ &= d_{ave_k} + \sum_{i=1}^n s_{(k,p_i)} \Delta p_i + N(0, \sigma_{rnd_k}^2) \end{aligned} \quad (2.15)$$

d_{ave_k} , $s_{(k,p_i)}$, σ_{rnd_k} は，統計的 MAX 演算後の最大パス遅延分布におけるパス遅延平均値，パス遅延感度，局所ばらつきの標準偏差である．ここで，全てのデバイスパラメータを正規分布に従うと仮定し，その平均と標準偏差をそ

れぞれ μ_i , σ_i とする. σ_i はデバイスパラメータ推定のばらつきと考えることができる. よって, 式 (2.15) の右辺第 2, 3 項は全て正規分布であるため, 正規分布同士の和となり, d_k は次のような正規分布として表現できる.

$$d_k \sim N\left(\sum_{i=1}^n (s_{(k,p_i)} \mu_i) + d_{ave_k}, \sum_{i=1}^n (s_{(k,p_i)} \sigma_i)^2 + \sigma_{rnd_k}^2\right) \quad (2.16)$$

ここで, d_k の平均と標準偏差をそれぞれ M_k , S_k すると, d_k の確率密度関数は次のようになる.

$$f(d_k|M_k, S_k) = \frac{1}{\sqrt{2\pi S_k^2}} \exp\left(-\frac{(d_k - M_k)^2}{2S_k^2}\right) \quad (2.17)$$

全テストパターンに対して, d_k , M_k , S_k を考えた時, これらは l 個の要素からなるベクトルであるため, それぞれ \mathbf{d} , \mathbf{M} , \mathbf{S} とする. よって, 全 l パターンに対する F_{\max} シグネチャ \mathbf{d} が得られる確率は次のようになる.

$$f(\mathbf{d}|\mathbf{M}, \mathbf{S}) = \prod_{k=1}^l \left\{ \frac{1}{\sqrt{2\pi S_k^2}} \exp\left(-\frac{(d_k - M_k)^2}{2S_k^2}\right) \right\} \quad (2.18)$$

式 (2.18) を尤度関数と考え, これを $L(\mathbf{M}, \mathbf{S})$ とする.

次に, $L(\mathbf{M}, \mathbf{S})$ を最大化する μ_i と σ_i を計算する. 正規分布における最尤推定法の計算では, 対数尤度関数を最大化する手法がしばしば用いられる.

$$\begin{aligned} \log\{L(\mathbf{M}, \mathbf{S})\} &= \log \prod_{k=1}^l \left\{ \frac{1}{\sqrt{2\pi S_k^2}} \exp\left(-\frac{(d_k - M_k)^2}{2S_k^2}\right) \right\} \\ &= \sum_{k=1}^l \left\{ \log \frac{1}{\sqrt{2\pi S_k^2}} - \frac{(d_k - M_k)^2}{2S_k^2} \right\} \end{aligned} \quad (2.19)$$

ただし, $M_k = \sum_{i=1}^n (s_{(k,p_i)} \mu_i) + d_{ave_k}$ であり, $S_k^2 = \sum_{i=1}^n (s_{(k,p_i)} \sigma_i)^2 + \sigma_{rnd_k}^2$ である. SSTA ツールを用いて, $s_{(k,p_i)}$, d_{ave_k} , σ_{rnd_k} を事前に計算可能である. これらを式 (2.19) に代入して, 全ての i に対する μ_i , σ_i を求める. μ_i と σ_i は, 数値計算パッケージの最適化関数を用いることで容易に導出できる.

2.3.3 F_{\max} シグネチャを用いたデバイスパラメータ推定実験

本推定手法により, デバイスパラメータを推定できることを確認するため, ISCAS'89 ベンチマーク回路 [52] を用いたシミュレーション実験を行う

表 2.2: 推定する ΔV_{thn} と ΔV_{thp} の組み合わせ (mV)

Case	1	2	3	4	5	6	7	8	9	10
ΔV_{thn}	41	-34	45	-25	76	-68	-53	1	74	-51
ΔV_{thp}	-46	10	13	-65	-20	-64	8	-66	-11	-1

た．本節では，商用の 65-nm プロセスの標準セルライブラリを用いて設計した回路 s38584 に対する適用結果を示す．

実験準備

市販 ATPG ツールを用いて，遷移遅延故障テストパターンを生成した [53]．テストパターン数は 1,833 で，故障検出率は 75.64 % である．SSTA を用いたパス抽出では，市販 STA ツールを用いて，119,228 パス抽出した [58]．この内，パス遅延故障シミュレーションにおいてロバスト条件で検出できたパスは 3,549 パスである．

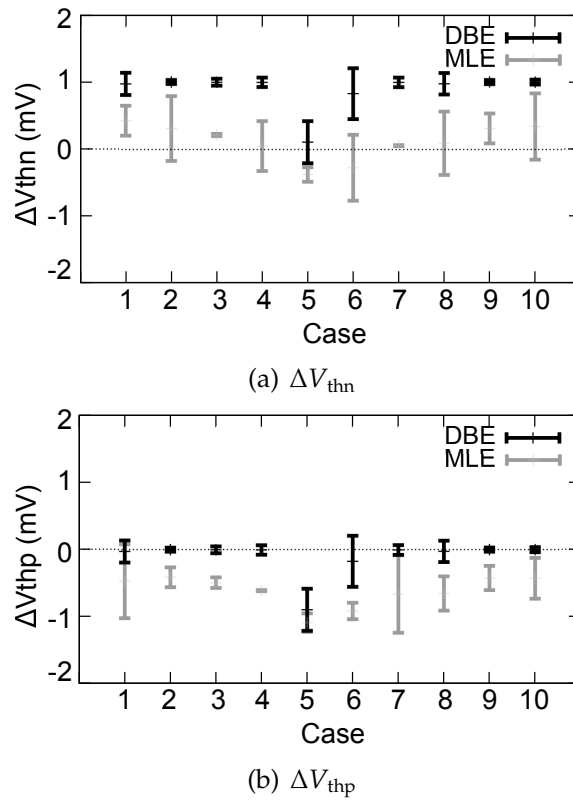
各推定法では，SSTA ツールを用いる代わりに，SPICE による Monte Carlo 回路シミュレーションを行った [54]．プロセスばらつきは，nMOS トランジスタと pMOS トランジスタのしきい値電圧とした．両者の分布は独立と仮定する．文献 [59] において，65-nm プロセスの局所ばらつき成分は大域ばらつき成分の 60% の大きさであることが報告されている．局所ばらつき成分はこれに従って設定した．一方，nMOS および pMOS トランジスタのデバイスパラメータを ΔV_{thn} および ΔV_{thp} とし，それぞれの範囲を -80 mV から $+80 \text{ mV}$ とする．文献 [60] から， F_{max} シグネチャを求める際の線形探索における探索分解能は 5.2 ps とする．

DBE 方式推定では，プロセスばらつき空間を，それぞれ 1 mV 単位で 25,921 個の小領域に分割している．また， $P(x)$ の初期事前分布は，一様分布とした．MLE 方式推定における式 (2.19) の計算は，R の *optim* 関数の L-BFGS (Limited-Broyden-Fletcher-Goldfarb-Shanno) アルゴリズムを用いた [61]．

デバイスパラメータ推定結果

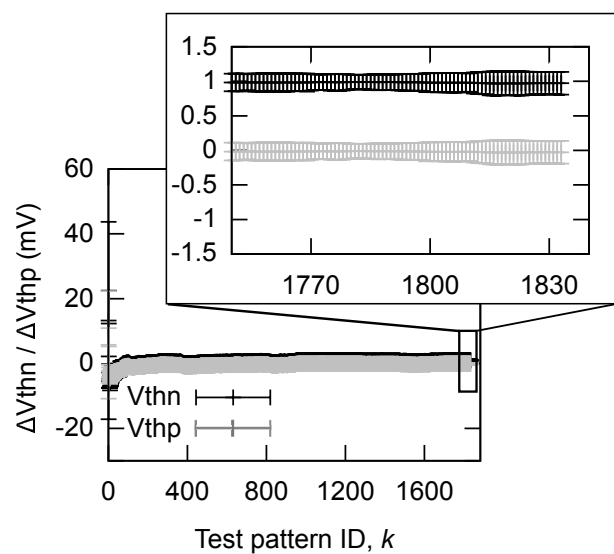
まず， -80 mV から 80 mV の間でランダムに生成した ΔV_{thn} と ΔV_{thp} の組み合わせ 10 組に対して，提案する 2 手法を適用した．表 2.2 に 10 組の組み合わせを示す．

表 2.2 の組み合わせに対して 2 手法を適用した結果を図 2.23(a)，2.23(b)

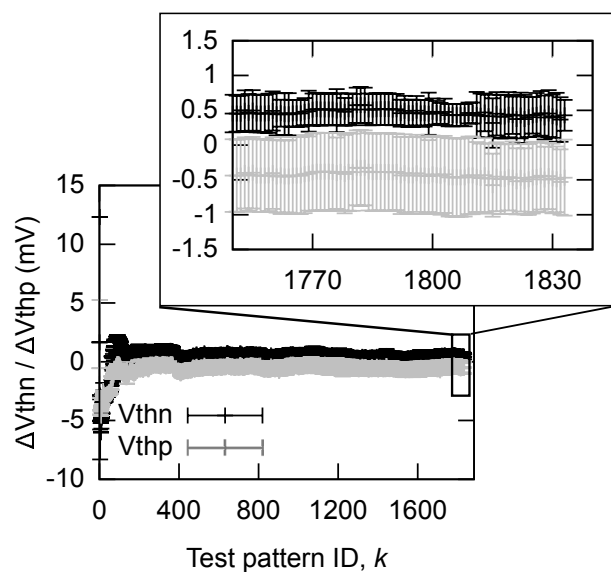
図 2.23: 10 点に対する ΔV_{thn} と ΔV_{thp} の推定結果

に示す。図 2.23(a), 2.23(b) は、それぞれ、 $k = 1,833$ の時の ΔV_{thn} と ΔV_{thp} の推定結果である。各図において、DBE 方式推定と MLE 方式推定の結果を示す。横軸は、各組み合わせのケース番号で、縦軸は、それぞれ ΔV_{thn} と ΔV_{thp} の推定誤差を表す。誤差棒の中心値は、正解領域と推定した ΔV_{thn} と ΔV_{thp} の平均値の差である。よって、中心値が 0 mV に近いほど、精度よく推定できていることを示す。誤差棒の縦軸方向は ΔV_{thn} と ΔV_{thp} の推定ばらつきの標準偏差を表している。すなわち、縦軸方向に長いほど、推定結果のばらつきが大きい。DBE 方式推定は、デバイスパラメータとして小領域毎の生起確率を推定する。これらに対し、 ΔV_{thn} と ΔV_{thp} の平均と標準偏差を計算して示している。MLE 方式推定では、推定した ΔV_{thn} と ΔV_{thp} の平均と標準偏差を記載している。図 2.23(a), 2.23(b) から、両手法とも、2 mV 以内でよく推定できていることが分かる。また、誤差棒の縦軸方向から、推定ばらつきの範囲は 1 mV 以内に収まっていることが分かる。

図 2.24(a), 2.24(b) に、ケース 1 の場合のテストパターン k に対する DBE 方



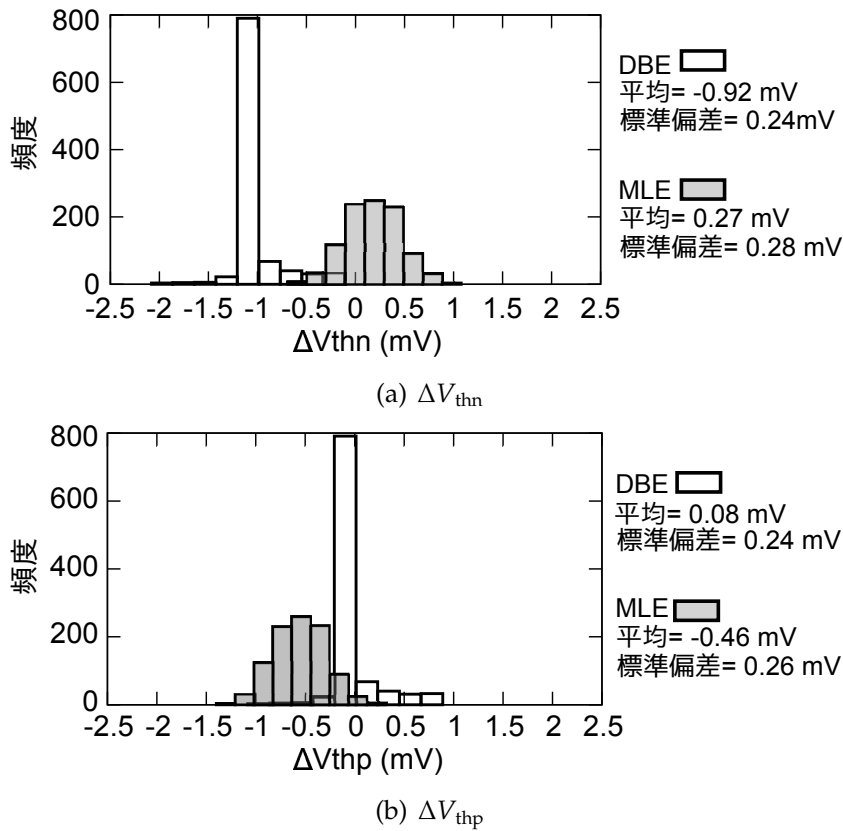
(a) DBE



(b) MLE

図 2.24: DBE 方式推定と MLE 方式推定におけるテストパターンに対する ΔV_{thn} と ΔV_{thp} の推定推移

式推定と MLE 方式推定の推定の推移を示す。各図において、 ΔV_{thn} と ΔV_{thp} の推定推移を示す。両手法とも、尤度関数を用いた推定を行っており、テストパターン k が増加することで、推定結果は高精度になることが期待される。横

図 2.25: 1,000 点に対する ΔV_{thn} と ΔV_{thp} の推定誤差のヒストグラム

軸はテストパターン ID, 縦軸は, ΔV_{thn} と ΔV_{thp} の推定誤差を示し, 図 2.23(a), 2.23(b) 同様, 誤差棒で表示している. 両手法とも, 最初の数パターンでは, 推定誤差が 40 mV あるいは 10 mV と非常に大きな値となっている. しかし, テストパターン k が増加することで, 急速に正解領域付近に収束していることが分かる.

続いて, -80 mV から 80 mV の間でランダムに生成した ΔV_{thn} と ΔV_{thp} の組み合わせ 1,000 組に対して 2 手法を適用した結果を示す. 図 2.25(a), 2.25(b) に, ΔV_{thn} と ΔV_{thp} の推定誤差のヒストグラムを示す. ここでは, 推定結果の標準偏差は用いず, 平均のみのヒストグラムとしている. 図 2.25(a), 2.25(b) から, ΔV_{thn} , ΔV_{thp} とともに推定誤差 3 mV 以内でよく推定できていることが分かる.

DBE 方式推定の適用結果において, ΔV_{thn} の推定誤差ヒストグラムの平均と標準偏差はそれぞれ -0.92 mV , 0.24 mV であり, ΔV_{thp} の誤差ヒストグ

ラムの平均と標準偏差はそれぞれ 0.08 mV , 0.24 mV である。一方で, MLE 方式推定の適用結果では, ΔV_{thn} の推定誤差ヒストグラムの平均と標準偏差はそれぞれ 0.28 mV , 0.28 mV であり, ΔV_{thp} の誤差ヒストグラムの平均と標準偏差はそれぞれ -0.46 mV , 0.26 mV である。ここで, いずれの手法においても, ΔV_{thn} と ΔV_{thp} の平均値が 0 mV となっていない。これは, F_{max} シグネチャを求める際の線形探索の分解能のためである。探索の分解能をより小さくすることで, これらの平均値を 0 mV に近づけることができることを実験的に確認している。

2.3.4 まとめ

本節では, F_{max} テストの枠組みを用いたデバイスパラメータ推定手法を提案した。提案手法では, テストパターン毎の F_{max} 値 (F_{max} シグネチャ) と, F_{max} テストで活性化されるパスの遅延分布を用いてデバイスパラメータを推定する。推定の方法として, 離散ベイズ推定法を用いる DBE 方式推定手法と, 最尤推定法を用いる MLE 方式推定手法の2手法を提案した。DBE 方式推定手法は, デバイスパラメータ空間を小領域に分割し, 推定対象チップが小領域に属する確率を計算する。パス遅延分布を尤度関数として, 小領域毎の確率を計算する。確率値が最大となる小領域を, そのチップのデバイスパラメータ値とする。MLE 方式推定手法では, デバイスパラメータ空間を分割せず, デバイスパラメータ値が正規分布に従ってばらつくと仮定する。MLE 方式推定手法では, F_{max} シグネチャが得られる確率を尤度としてデバイスパラメータを推定する。両手法とも, F_{max} テストから得られる情報と SSTA で計算される情報を用いているため, 提案手法のための特別な回路は不要である。計算機実験では, DBE 方式推定ではデバイスパラメータ空間を 1 mV で分割した。提案手法の評価により, トランジスタのしきい値の大域ばらつき成分を, 3 mV の精度で推定できることを示した。

2.4 まとめ

本章では，テスト結果を用いたデバイスパラメータ推定手法を提案した．本論文では，デバイスパラメータ推定に基づく適応型テストを提案しており，本推定手法は，適応型テストの効果を決定する重要な工程である．本章では，2種類の推定手法を提案した．1つ目の手法は，IDDQ測定により得られるIDDQシグネチャを用いる．65-nmプロセスを用いた計算機実験において，デバイスパラメータ空間を5mV刻みで分割し，トランジスタのしきい値電圧のばらつきを5mV精度で推定できることを示した．2つ目の手法は， F_{\max} テストの枠組みを用いる．パラメータ推定の方法として離散ベイズ推定法，最尤推定法を用いる2種類の手法を提案した．離散ベイズ推定法を用いた手法では，デバイスパラメータ空間を小領域に分割し，小領域毎の生起確率を導出する．計算機実験において，最尤推定法を用いた手法では，1mV刻みで小領域を構成した．離散ベイズ推定を用いた方式，最尤推定法を用いた方式いずれの手法においても，トランジスタのしきい値電圧を3mV以内の精度で推定できることを示した．これら2つの推定手法は同時に適用可能であるため，両手法の推定結果を比較することで，高信頼な特性推定結果を得ることができる．

チップ面積が大きい場合は，局所ばらつきが大きくなることが知られている [19]．本章で提案したデバイスパラメータ推定手法は，大域ばらつきのみを考慮しているため，推定精度が劣化する可能性がある．今後，チップ面積が大きいチップにおいても高い推定精度を確保するため，システムティックな局所ばらつき成分をモデル化し，本推定手法に追加する必要がある．

第3章

特性推定に基づく適応型IDDQテスト良否判定基準決定

本章では，デバイスパラメータ推定に基づいて IDDQ テストの良否判定基準を決定する手法を提案する．第2章で提案した IDDQ シグネチャを用いたデバイスパラメータ推定手法により，DUT 毎にデバイスパラメータを推定する．パラメータ推定結果から，DUT 毎のリーク電流値の上限値を計算する．これを基に，DUT 毎の IDDQ テスト良否判定基準を決定する．本手法を適用することでチップ毎に最適な良否判定基準を設定でき，故障見逃しと歩留まり損失を低減し，テスト品質を向上する．

3.1 はじめに

文献 [30] では，遅延ばらつきに比べてリーク電流ばらつきの方が大きく，パス遅延値が製品規格内であったとしてもリーク電流規格に収まらないチップが多数あることを報告している．このような大リーク電流下での IDDQ テストの良否判定は困難を極める．IDDQ テストは，ショート故障やオープン故障などの物理欠陥に対する検出能力が非常に高く，広く量産テストに用いられてきた．しかし，微細プロセスにおいては，特性ばらつきの増大によりリーク電流ばらつきが増加し，IDDQ テストの良否判定基準の設定が困難になっている．

特性ばらつきを考慮した IDDQ テスト良否判定基準設定の容易化に関する研究は多く行われている [62–67]．文献 [62, 63] では， k 平均法 [64] などのクラスタリングアルゴリズムを用いて IDDQ テストの良品と不良品の判定を行う．文献 [65, 66] では，IDDQ 測定値と他のテスト項目の測定値を用いた多変量解析によるテスト手法が提案されている．また，文献 [67] では IDDQ テスト良否判定基準をウェハ毎，ロット毎に変更する手法が提案され

ている。しかし、これらの手法は、最適な良否判定基準を得るために、多くの IDDQ 測定データが必要となり、少量生産の LSI には適用が困難である。

本章では、デバイスパラメータ推定に基づいて IDDQ テストの良否判定基準を決定する手法を提案する。本手法では、IDDQ シグネチャを用いたデバイスパラメータ推定手法（2.2 節）を用いて、最適な良否判定基準を DUT 毎に適応的に設定する。同様に、 F_{\max} テストの枠組みを用いた推定手法（2.3 節）も適用可能である。本章で提案する良否判定基準決定法は、計算時間が長いという課題がある。そこで、この計算を適用する前に、クラスタリングアルゴリズムを用いて、大きなリーク電流を発生させる故障チップを検出する。これにより良否判定基準計算が適用されるチップ数を削減し、全体の計算時間を削減する。本章では、デバイスパラメータ推定に基づく IDDQ 良否判定基準計算を含む、2 段階の IDDQ テスト手法を提案する。第 1 段階では、クラスタリングアルゴリズムによる良品と不良品の判定を行い、第 2 段階では、2.2 節で提案したデバイスパラメータ推定手法を用いて、良品と不良品の良否判定基準を決定する。

以下、本章は次のように構成される。3.2 節にて、2 段階の IDDQ テスト手法を提案する。続く 3.3 節で、ISCAS'89 ベンチマーク回路の s38584 を用いた計算機実験結果を示し、最後に 3.4 節で本章をまとめる。

3.2 2 段階 IDDQ テスト手法

3.2.1 基本概念

図 3.1 に、提案する 2 段階手法の基本概念を示す。まず、第 1 段階において、クラスタリングアルゴリズムを用いて、IDDQ 電流値が大きな DUT を異常チップとして不良判定する。第 1 段階で不良と判定されなかったチップが続く第 2 段階のテストに適用される。第 2 段階においては、デバイスパラメータ推定に基づくテスト良否判定基準決定手法が適用される。第 1 段階で故障リーク電流の大きな DUT を不良判定することで、第 2 段階が適用されるチップ数が削減され、全体の計算時間を削減できる。

2.2 節で提案したデバイスパラメータ推定手法を適応型テストに適用するための課題として、推定に用いるチップが不良を含んでいる可能性を考慮しなければならない。物理欠陥が存在した場合、IDDQ 電流シグネチャは故障の影響により変調する。リーク故障を含む IDDQ 電流シグネチャを用いてデバイスパラメータ推定を行った場合、推定精度が劣化してしまう。そこ

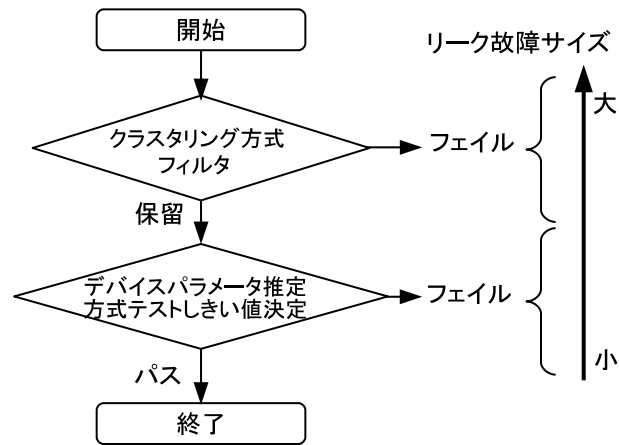


図 3.1: 2 段階 IDDQ テスト手法の概念図

で，提案する IDDQ テスト手法では，IDDQ シグネチャに故障の影響があったとしても正しくデバイスパラメータを推定するために，故障活性ベクトルを新たに導入する．故障活性ベクトルを用いて，IDDQ 電流シグネチャ内で故障の影響の有無を推定する．故障活性ベクトルの計算には，焼きなまし (SA) 法を用いる [68]．

しかし，SA 法における計算量が非常に大きいため，IDDQ 良否判定基準決定に長い計算時間を要する．よって，全 DUT に対する計算時間を削減するために，SA 法を適用する DUT の数を削減する．提案手法では，図 3.1 に示すように，リーク故障電流が大きい故障の DUT を，第 1 段階のクラスタリング方式フィルタによってテストすることで，SA 法が適用されるチップ数を削減する．

以下，本章では，不良チップは単一縮退故障を含むと仮定する．

3.2.2 クラスタリング方式フィルタ

提案する 2 段階 IDDQ テストでは，まず，クラスタリング方式フィルタによって，リーク故障の大きな DUT を検出する．文献 [29] では，不良チップの IDDQ シグネチャの IDDQ 電流値は故障の影響で電流量が増加されることが報告されている．クラスタリング方式フィルタでは，DUT 毎の IDDQ 電流シグネチャに対して，クラスタリングアルゴリズムを適用することで，低 IDDQ 電流値のクラスタ C_L と高 IDDQ 電流値のクラスタ C_H に分類する．DUT に故障が存在する場合， C_H には故障の影響によって活性化された

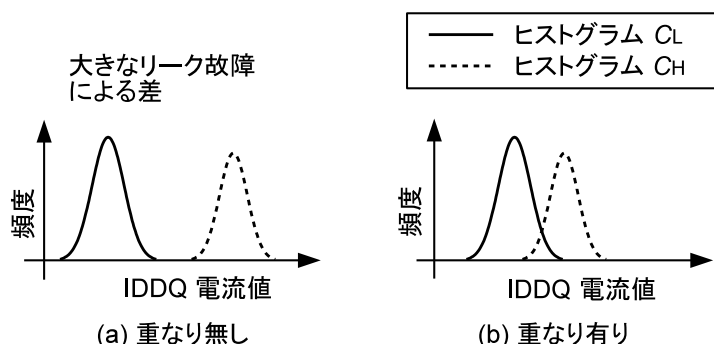


図 3.2: クラスタリング方式フィルタの概念図

故障 IDDQ 電流値が含まれるはずである。

図 3.2 にクラスタリング方式フィルタの概念図を示す。図 3.2 では、クラスタ C_L と C_H に対応するヒストグラムを示す。サイズの大きなリーク故障電流が存在した場合、図 3.2(a) のように、これらのヒストグラムの間には大きな差が存在する。反対に、故障が存在しない、あるいは故障サイズが小さい場合は、図 3.2(b) のようにヒストグラム同士が重なり合う。図 3.2(a) の場合は、明らかに 2 ヒストグラムが故障の影響を含むものと含まないものに分類されていることが分かり、容易に故障の有無を判定することができる。しかし、図 3.2(b) の場合は、故障の影響ではなく、プロセスばらつきの影響による IDDQ 電流値の増減、または IDDQ テストパターン毎の回路状態の差による増減の可能性もあるため、不良品と判定することは難しい。このような IDDQ 電流シグネチャの場合、第 1 段階では保留とし第 2 段階で詳細にテストされる。

IDDQ シグネチャを構成する IDDQ 電流値を C_L と C_H の 2 クラスタに分類するため、提案手法では、 k 平均法アルゴリズムを適用する [64]。 k 平均法アルゴリズムでは、IDDQ シグネチャを入力とし、IDDQ シグネチャ内の IDDQ 電流値を 2 クラスタ C_L と C_H に分類する。ここで、測定 IDDQ シグネチャを構成する IDDQ 電流値を I_i とする。 I_i は i 番目のテストパターンによって得られる IDDQ 電流値である。 k 平均法で使用される距離は、ユークリッド距離とする。 k 平均法は次のように適用される。

1. I_i を 2 クラスタ C_L と C_H にランダムに振り分ける。
2. 両クラスタの中心を計算する。ここで、クラスタの中心はクラスタ内の電流値の平均と定義する。
3. 電流値 I_i と中心との距離を計算し、中心が最も近いクラスタに振り分

け直す。

4. ステップ 2 と 3 を繰り返し、クラスタ内の電流値に変化がなくなれば終了する。

クラスタリング後、クラスタ間に十分大きな差が存在するか否かを自動で判定する必要がある。この自動判定のために、シルエットプロットを用いる [69]。シルエットプロットは、クラスタリングアルゴリズムの評価に有用な尺度である。シルエット値 $s(i)$ はクラスタ内の各電流値に対して計算され、自値と他クラスタがどれだけ離れているかを評価する数値である。シルエット値 $s(i)$ は次のように定義される。

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (3.1)$$

ここで、 $a(i)$ は i 番目の電流値 I_i における自クラスタ内の他電流値との距離の平均で、 $b(i)$ は i 番目の電流値 I_i における他クラスタ内の電流値との距離の平均である。式 (3.1) から、 $s(i)$ は -1 から 1 の間の数値をとる。 $s(i)$ が大きな値をとる場合、 i 番目の電流値と他クラスタは十分離れていることを示唆する。ここで、 $s(i)$ においても、十分離れているか否かのしきい値を設定しなければならず、このしきい値は十分大きな値とする。このしきい値を超えない場合、クラスタリング方式フィルタでは不良品と判定せず、次段階におけるデバイスパラメータ推定に基づくテスト良否判定基準決定においてテストされる。

3.2.3 デバイスパラメータ推定に基づく IDDQ テスト良否判定基準決定

第 2 段階では、第 2 章で提案したデバイスパラメータ推定に基づいて、IDDQ 電流テストの良否判定基準を決定する。DUT 毎のデバイスパラメータ値を推定した後、DUT に対する IDDQ 電流分布を計算し、これを基に IDDQ 電流テストしきい値を計算する。

第 2 章においては、良品と判定されたチップの IDDQ 電流シグネチャを対象としていたが、実際のテスト環境においては、テスト対象チップに故障がある可能性がある。故障のあるチップの IDDQ シグネチャを用いてデバイスパラメータ推定を行った場合、推定精度が劣化する。よって、故障の影響を含む IDDQ シグネチャを用いても正しくデバイスパラメータを推定する必要がある。これは、テストによる良否判定を行う前に、推定のために IDDQ

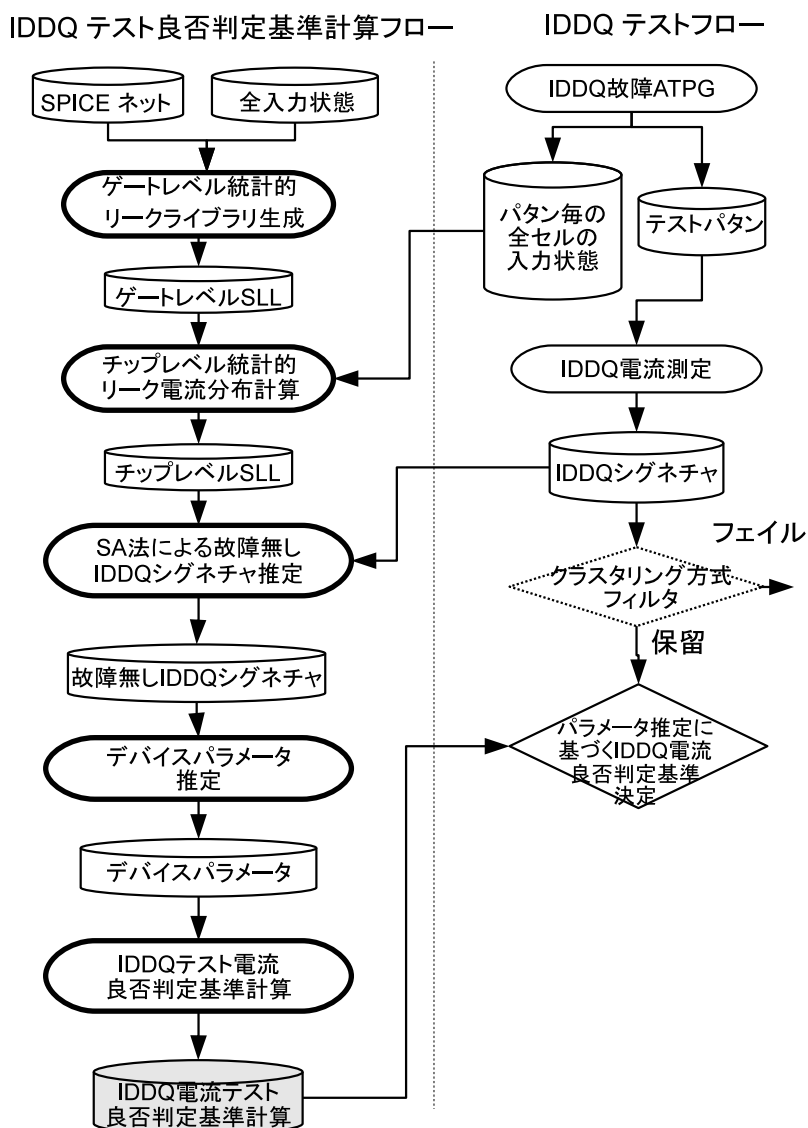


図 3.3: デバイスパラメータ推定に基づく IDDQ テスト良否判定基準決定の全体フロー

シグネチャ内の故障の有無を判定することを意味する。本テスト手法では、この課題を SA 法を用いて解決する。SA 法を用いて故障無し IDDQ シグネチャを推定し、このシグネチャを用いてデバイスパラメータを推定する。最後に、推定したデバイスパラメータ値を基に IDDQ 電流テスト良否判定基準を決定する。

デバイスパラメータ推定に基づくテスト良否判定基準決定の全体フロー

を図 3.3 に示す。図 3.3 では、第 1 段階のクラスタリング方式フィルタも含む。IDDQ テストフローにおいて、自動テストパターン生成 (ATPG) ツールを用いて、IDDQ テスト用のテストパターンを生成し、同時にテストパターン毎の回路内部状態を得る。生成したテストパターンを用いて、テスト対象チップの IDDQ 電流シグネチャを測定する。デバイスパラメータ推定に基づく IDDQ 良否判定基準の決定は次の 5 工程で構成される。

1. ゲートレベル統計的リーク電流ライブラリの作成
2. チップレベル統計的リーク電流分布の計算
3. SA 法による故障無しシグネチャの推定
4. 故障無しシグネチャを用いたデバイスパラメータ推定
5. IDDQ 電流テストの良否判定基準の決定

ゲートレベル統計的リーク電流ライブラリの作成

2.2.1 節同様、与えられた標準ロジックセルに対して、統計的リーク電流ライブラリ (ゲートレベル SLL) を生成する。まず、想定するデバイスパラメータの変動空間を小領域に分割する。ゲートレベル SLL は、全セルの小領域における、各セルの入力状態毎の統計的リーク電流分布を計算したものである。標準ロジックセルのリーク電流分布は、ばらつきパラメータのデバイスパラメータ値を各領域の値 (大域ばらつき) に固定し、パラメータの局所ばらつきの分布にしたがう乱数値を加えたネットリストを作成して、Monte Carlo 回路シミュレーションにより計算する。

チップレベル統計的リーク電流分布の計算

チップレベル SLL として、テスト対象回路の統計的リーク電流分布を計算する。チップレベル SLL は、全小領域と全テストパタンの組合せに対して行う。テスト対象回路のリーク電流分布は、各セルのリーク電流分布の統計的総和演算によって得られる [30,32,51]。IDDQ テストパタンの生成時に回路中の全セルの入力状態が得られるため、ゲートレベル SLL を参照することでテスト対象回路の統計的リーク分布を計算できる。

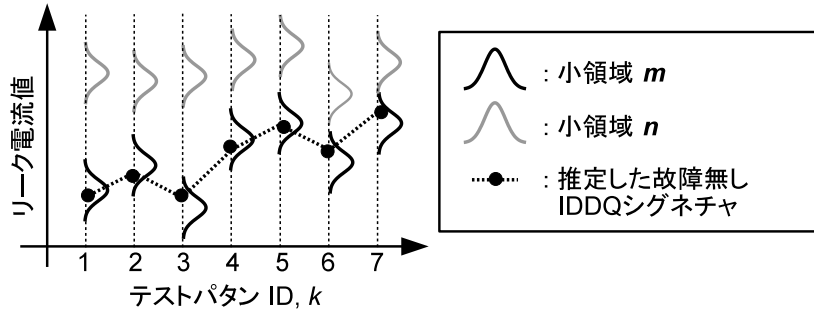


図 3.4: IDDQ 電流値の統計的リーク電流分布に対する当てはまりの良さの概念図

故障無し IDDQ シグネチャの推定

本工程では、チップを測定して得られた IDDQ シグネチャから故障無し IDDQ シグネチャを推定する。この工程は、故障の有無に関わらず、デバイスパラメータを正確に推定するために行う。故障無し IDDQ シグネチャを得るために、まず、与えられたテストパターンによって故障が活性化されたか否かを判定する必要がある。これを故障活性化ベクトル \mathbf{v} とし、その要素は対応するテストパターンが故障を活性化するか否かを表す。テストパターン i が故障を活性化する場合 $v_i = 1$ 、活性化しない場合 $v_i = -1$ とする。故障活性化ベクトルの要素数は、テストパターン長に等しい。ここでは、故障活性化ベクトルを求める最適化問題に定式化し、SA 法により求める。

まず、図 3.4 を用いて、最適化問題のコスト関数について考える。この例では、7 個の IDDQ 電流値がある推定故障無し IDDQ 電流シグネチャと 2 つの小領域 m と n から得られるチップレベル統計的リーク電流分布を考える。図 3.4 に示すように、真の故障活性化ベクトルにより故障無し IDDQ シグネチャが得られた場合、全てのテストパターンにおける IDDQ 電流値は、ある小領域における統計的リーク電流値とよく当てはまるはずである。リーク電流分布は、チップレベル SLL より得られる。この例では、推定故障無し IDDQ 電流シグネチャは、小領域 m の方がよく当てはまる。ここで、コスト関数 OPT を次のように定義する。

$$OPT = \max_x (\min_i (L_i(\mathbf{x}|I'_i))) \quad (3.2)$$

I'_i は推定された i 番目の故障無し IDDQ 電流値である。 $L_i(\mathbf{x}|I'_i)$ は、小領域 \mathbf{x} のチップレベル SLL に対する I'_i の尤度であり、 I'_i の小領域 \mathbf{x} への当てはま

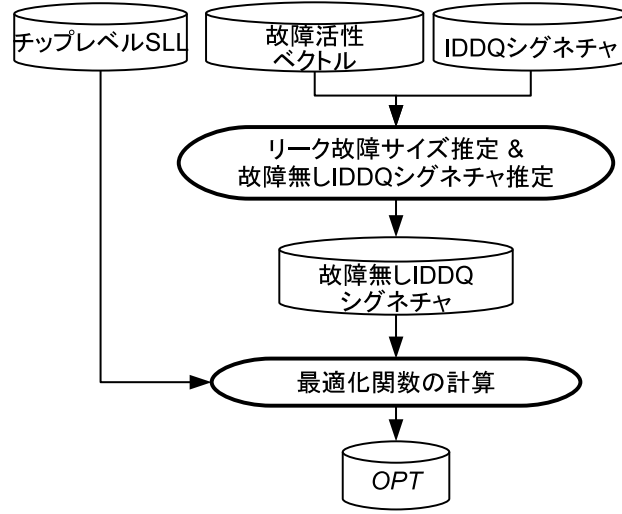


図 3.5: 最適化関数計算の処理

り具合を評価する尺度である．尤度 $L_i(\mathbf{x}|I'_i)$ は次のように計算される．

$$L_i(\mathbf{x}|I'_i) = \frac{1}{\sqrt{2\pi}\sigma_{(x,i)}I'_i} \exp\left(-\frac{(\ln I'_i - \mu_{(x,i)})^2}{2\sigma_{(x,i)}^2}\right) \quad (3.3)$$

ここで、 $\mu_{(x,i)}$ と $\sigma_{(x,i)}$ は、小領域 \mathbf{x} における i 番目のテストパタンの時の統計的リーク電流分布の対数値の平均と標準偏差である． $\mu_{(x,i)}$ と $\sigma_{(x,i)}$ はチップレベル SLL から得られる．図 3.4 の \mathbf{m} のような状況を考えると、特定の小領域に対してのみ全ての i に対して、尤度が高い必要がある．式 (3.2) では、まず、小領域毎の全 i に対する尤度の最小値を求める．そして、最小値を最大化する領域を求めて、その時の尤度を OPT とする．SA 法では、 OPT を最大化するように故障活性化ベクトルを計算する．

図 3.5 に、SA 法における OPT を計算する一連の処理を示す．この処理の入力は、測定した IDDQ 電流シグネチャと故障活性ベクトル \mathbf{v} である．故障活性ベクトルの数値は、SA 法においてランダムに生成される．上記の入力に対して、まず、リーク故障のサイズが推定される．もし、 i 番目の IDDQ 電流値が故障を含む場合、すなわち $v_i = -1$ の時、推定 IDDQ 電流値 I'_i は $I_i - \delta$ になる． δ はリーク故障の大きさとする．リーク故障サイズ δ は、故障活性した IDDQ 電流値の平均値と活性化していない電流値の平均値の差として、簡易に計算する．一方で、 $v_i = 1$ の時、 I'_i は I_i に等しい．このように、ランダムに決定された故障活性ベクトル \mathbf{v}_i から I'_i を計算し、式 (3.2) を用いて OPT を計算する．

SA 法で得られる解は準最適解であるため、故障活性ベクトルの要素に 1 (故障が活性化されたことを表す) があつたとしても、チップを不良判定することはできない。そこで、従来手法では故障無し IDDQ シグネチャ推定時に同時に得られるデバイスパラメータ推定値を用いて IDDQ 電流良否判定基準を計算し、これを用いて故障の有無を判定する。

故障無し IDDQ シグネチャを用いたデバイスパラメータ推定

本工程では、先の工程で推定した故障無し IDDQ 電流シグネチャを用いてデバイスパラメータを推定する。デバイスパラメータ推定は、2.2 節で提案した推定手法を用いる。小領域 \mathbf{x} の生起確率を事前確率として、IDDQ 値から、あるチップが小領域 \mathbf{x} に属する尤度を計算する。ベイズの定理より、事後分布 $P(\mathbf{x}|I'_i)$ は次のように得られる。

$$P(\mathbf{x}|I'_i) = \frac{L_i(\mathbf{x}|I'_i)P(\mathbf{x}|I'_{i-1})}{C} \quad (3.4)$$

ここで、 C は $\sum_{\mathbf{x}} P(\mathbf{x}|I'_i) = 1$ を満たすような正規化定数であり、 $P(\mathbf{x}|I'_{i-1})$ は事前分布である。 $i = 1$ の時の事前分布は $P(\mathbf{x})$ とする。IDDQ 測定回数が增多することで、すなわち、テストパターン i が増加することで、IDDQ 電流値 I'_i が測定された後の事後分布が更新される。

IDDQ 良否判定基準の決定

本工程では、各テストパターン k 毎に対して、DUT 毎の良品と不良品の判定基準を計算する。判定基準は、その DUT の全小領域を考慮した統計的リーク電流分布から上限値を計算することで得る。今、前工程から、小領域毎の生起確率 $P(\mathbf{x}|I_k)$ と小領域 \mathbf{x} における統計的リーク電流分布 $f_{(\mathbf{x},k)}(I_k)$ を得ている。全小領域を考慮した統計的リーク電流分布は、各小領域の統計的リーク電流分布を各小領域の生起確率で重み付けして重ね合わせることで得られる。DUT の統計的リーク電流分布は、次に示す重み付け和で計算される。

一次元の場合の例を図 3.6 に示す。図 3.6(a) では、小領域毎の推定確率値が示されている。まず、小領域毎の統計的リーク電流分布に、小領域毎の生起確率で重み付けする。図 3.6(b) に示すように、各小領域における統計的リーク電流分布は、チップレベル SLL から得られる。DUT の統計的リーク電流分布は、図 3.6(c) のように、正規分布に近似することで得られる。

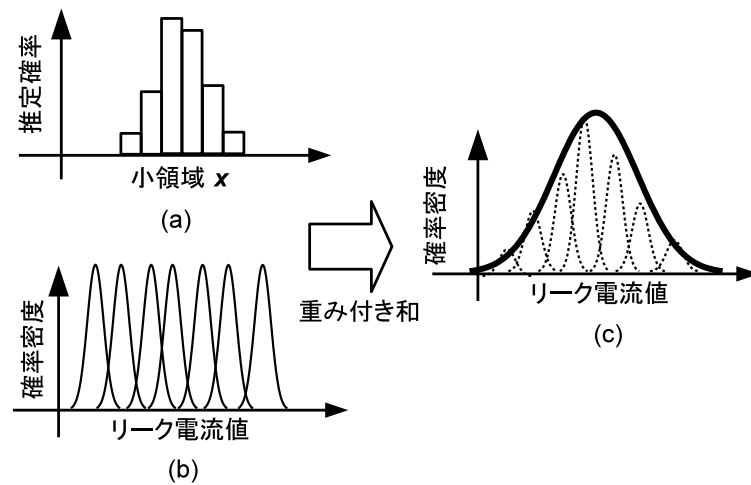


図 3.6: デバイスパラメータと局所ばらつきを考慮した統計的リーク電流分布の計算方法

例えば、測定 $IDDQ$ 値が計算した統計的リーク電流分布の 3σ を超えた場合を不良と判定することを考える．リーク電流分布の累積密度分布から 3σ における電流値は計算でき、測定 $IDDQ$ 値 I_i がこれを超えると不良と判定される． $IDDQ$ シグネチャの $IDDQ$ 値の情報を更新することで、生起確率 $P(x|I_k)$ が更新されるため、各プロセス条件の統計的リーク電流分布の重み付けが更新され、全デバイスパラメータ領域を考慮した統計的リーク電流分布も更新される．よって、 $IDDQ$ テストパターンが多いほど、 $IDDQ$ 良否判定基準は高精度化される．

3.3 シミュレーション実験

65-nm 標準セルライブラリを用いて設計する ISCAS'89 ベンチマーク回路 s38584 [52] に対し、提案する 2 段階 $IDDQ$ 手法の適用した．ここでは、提案手法によるテスト品質向上の効果を確認するため、次の 2 つの実験を行った．

従来法との比較

仮想ウェハ上に s38584 が製造されていることを想定し、文献 [33,34] で提案されている NNR 法と比較

リーク故障サイズ毎の検出能力の評価

提案手法におけるリーク故障サイズ毎の検出能力を評価

3.3.1 実験準備

提案手法と従来手法を C 言語で実装し、実験を行った計算機は Intel(R) Core(TM) i7-2600S (CPU: 2.8 GHz, キャッシュ: 8192 KB) である。全ての工程は 1 スレッドで実行した。

ゲートレベル SLL 生成において、リーク電流値は、SPICE シミュレータ [54] を用いた Monte Carlo シミュレーションを行った。nMOS と pMOS のデバイスパラメータをトランジスタのしきい値電圧のばらつきとした。しきい値電圧の局所ばらつき成分は、文献 [55] に従うとした。以下、本節では、nMOS と pMOS のしきい値電圧の大域ばらつき成分 (デバイスパラメータ) を ΔV_{thn} , ΔV_{thp} と表記する。想定するばらつきパラメータ空間は、 -80 mV から 80 mV とし、小領域は 10 mV 間隔とした。全小領域数は 289 個である。

テストパターン生成は、商用の ATPG ツールを用いた [53]。擬似縮退故障モデルを対象にテストパターンを生成し、テストパターン数は 49 で、故障検出率は 100 % である。

$P(x)$ の初期事前分布は一様分布とした。ここで、 x は ΔV_{thn} と ΔV_{thp} の 2 次元ベクトルである。模擬 DUT の IDDQ 電流シグネチャは、シミュレーションにより得た。シミュレーション中において、デバイスパラメータ値 ΔV_{thn} と ΔV_{thp} を決定し、この数値に対して局所ばらつきを付加して、IDDQ 電流値を計算している。

シルエットプロットが提案されている文献 [69] では、 $s(i)$ の平均が 0.74 であれば、クラスタ間が十分離れていることが報告されている。クラスタ間が十分離れているか否かのしきい値の決定において、楽観的なしきい値を設定してしまった場合、故障見逃しの要因となる。本実験では、悲観的な評価を行う。全要素の $s(i)$ の最小値が 0.74 を下回る場合、テスト対象チップは故障の疑いがある、として次段階において詳細にテストする。

以下、本節では、小領域を $(\Delta V_{\text{thn}}, \Delta V_{\text{thp}})$ のように括弧を用いて表記する。例えば、 $\Delta V_{\text{thn}} = 10 \text{ mV}$, $\Delta V_{\text{thp}} = -10 \text{ mV}$ の場合は、 $(-10 \text{ mV}, -10 \text{ mV})$ と表記する。

テスト精度を評価するため、本実験では歩留まり損失と故障見逃しを次のように定義する。

- 歩留まり損失:
全チップ数に占める'フェイル'と判定された良品チップ数の割合
- 故障見逃し:
全チップ数に占める'パス'と判定された不良品チップ数の割合

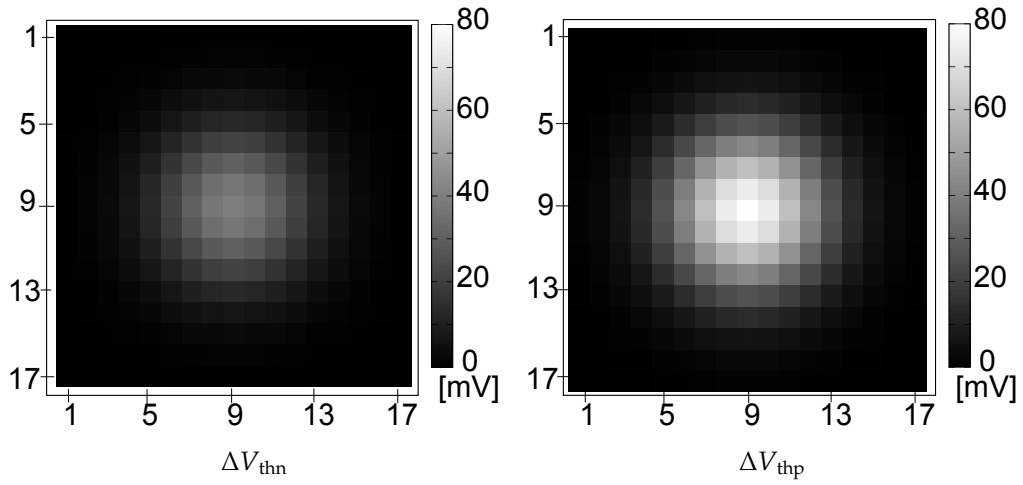
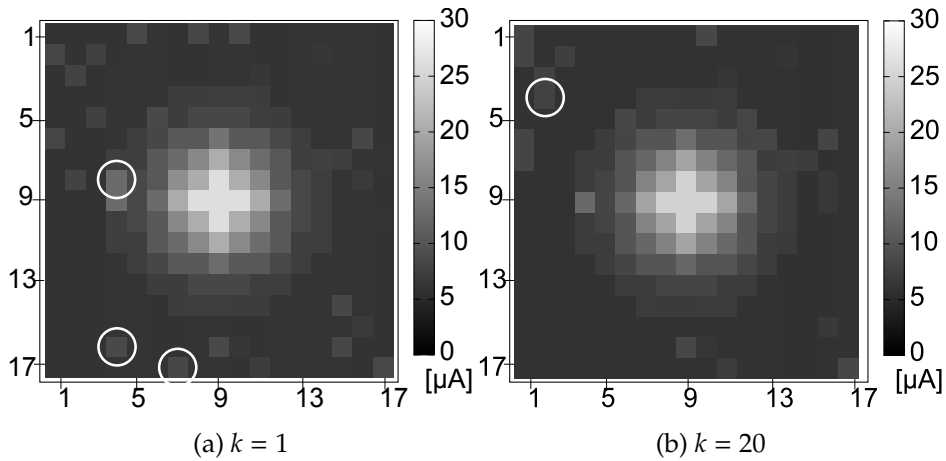


図 3.7: 仮想ウェハ上のデバイスパラメータの分布

3.3.2 従来手法との比較結果

図 3.8: 仮想ウェハ上の $k = 1$ と $k = 20$ の時の $IDDQ$ 電流分布

本実験では、仮想ウェハ上に s38584 が $17 \times 17 = 289$ チップ製造されていると仮定する。ウェハ上における ΔV_{thn} と ΔV_{thp} の分布を図 3.7 に示す。縦軸と横軸は、ウェハ上でのチップの座標を表しており、図内の四角形が s38584 の各チップに対応する。 ΔV_{thn} は中心が最も高い同心円状に分布しており、0 mV から 40 mV まで変化している。 ΔV_{thp} も同様の形状で分布しており、0 mV から 80 mV まで変化している。また、このウェハで製造されるチップの歩留まりを 80% とし、289 チップの内、良品が 231 チップで不良品が 58 チップあると仮定する。不良チップには、故障によるリーク故障があること

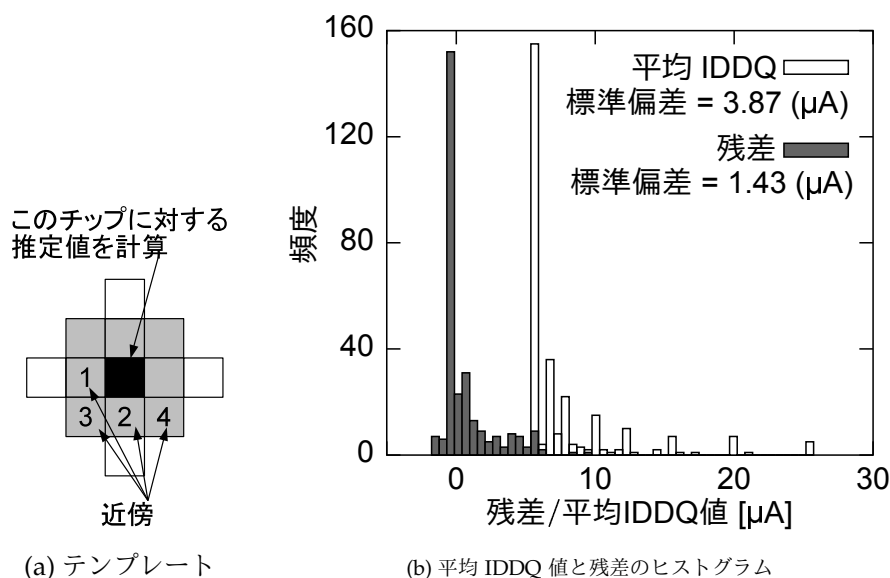


図 3.9: (a) 推定値計算のためのテンプレート. DUT の周辺 8 チップを推定値計算に使用する (b) 289 DUT に対する測定 IDDQ シグネチャの平均値と残差のヒストグラム.

を仮定し、信号線が VDD か GND にショートする単一縮退故障を想定する. 物理欠陥によるリーク電流のサイズ δ (μA) は、 $P(\lambda) = 0.45 \exp(-0.45\delta)$ に従う分布からランダムに選択する. 上記の仮定により、テストパターン $k = 1$, $k = 20$ の時の、ウェハ上の各チップにおける IDDQ 電流の分布を図 3.8 に示す. IDDQ 電流分布の傾向として、図 3.7 から、大域ばらつきの影響で同心円状に分布しているが、ランダムに発生する物理欠陥の影響でリーク電流値が高いチップがあることが分かる.

本実験では、従来法の 1 つである NNR 法との比較を行う. ここで NNR 法について簡単に説明する. NNR 法では、良品と不良品の判定に、真値と推定値の残差を用いる. 真値は、DUT の IDDQ シグネチャの平均 IDDQ 値である. 推定値は、DUT の近傍として選択されたチップの IDDQ シグネチャの平均値として定義され、次の 2 ステップで計算される.

1. まず、図 3.9(a) に示すようなテンプレートを定義する. 本実験では、DUT の周辺 8 チップ (グレー) をテンプレートとし、近傍の数は 4 とする.
2. テンプレート上のチップの測定 IDDQ シグネチャの真値が小さい順に 4 チップ (近傍数より決定) を選択し、これらの真値の平均を推定値とする. 図 3.9(a) の例では、1 から 4 の番号が記載されたチップが近

表 3.1: クラスタリング方式フィルタ結果

‘フェイル’判定された不良チップ	43.10 % (25/58)
‘保留’判定された良品チップ	0 % (0/231)

傍として選択されている。

図 3.9(b) に、289 DUT に対する平均 IDDQ 値と残差のヒストグラムを示す。図 3.9(b) から、平均 IDDQ 値の標準偏差は $3.87\mu\text{A}$ で残差の標準偏差は $1.43\mu\text{A}$ である。NNR 法は、ウェハー上の特性ばらつきは滑らかに変化していることを前提とした手法である。この前提が成り立つ場合、DUT 周辺の IDDQ 電流値と DUT の IDDQ 値が近い値となる。よって、例えば特性ばらつきが大きくとも故障の無い DUT の残差は小さい値となる。一方で、故障があった場合、DUT の残差は大きい値となる。文献 [33,34] で報告されているように、NNR 法を適用することで、平均 IDDQ 値よりも残差の方が標準偏差が小さくなっており、良品チップと不良品チップを切り分けやすくなる。

まず、表 3.1 にクラスタリング方式フィルタの結果を示す。不良チップのみが‘フェイル’として判定されており、良品チップは全て‘保留’と判定されている。‘保留’は、‘フェイル’判定されなかったことを示し、‘保留’の DUT は第 2 段階の良否判定基準計算で詳細にテストされる。表 3.1 から、クラスタリングアルゴリズムが有効に働いており、次段階の良否判定基準計算が適用されるチップ数は、261 チップまで削減されている。クラスタリング方式フィルタが適用されなかった場合、289 チップ全てが良否判定基準計算の対象となる。良否判定基準の計算は、1 チップあたり 306 分要し、クラスタリング方式フィルタを適用することで、全体で 7650 分削減できている。

図 3.10 に、(7mV, 14mV) のチップの IDDQ 電流値のヒストグラムを示す。このチップは、 $1.90\mu\text{A}$ の故障が存在する。図において、2つのヒストグラム C_L と C_H が k 平均法によって図 3.2(a) のように分類されていることがわかる。この例におけるシルエットプロットの分布を図 3.11 に示す。縦軸に C_L と C_H に分類した IDDQ シグネチャの全要素を表し、横軸に対応するシルエット値を示す。図 3.11 では、全ての要素においてシルエット値が 1.0 に近い数値となっており、クラスタリング方式フィルタにおけるしきい値である 0.74 を上回っていることが分かる。このように、リーク故障が大きいチップに対しては、クラスタリング方式フィルタによって検出できている。

表 3.2 に、提案する 2 段階 IDDQ テスト手法と NNR 法を適用した際のテスト精度をまとめる。両手法ともテストしきい値を 1σ から 9σ に変化させ

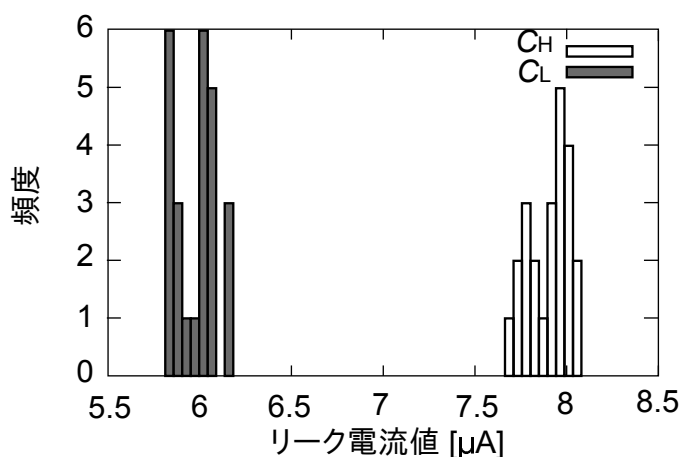


図 3.10: k 平均法によって分離された C_L と C_H のヒストグラム の例. この例においては, 2 つのヒストグラムが明確に分離されている.

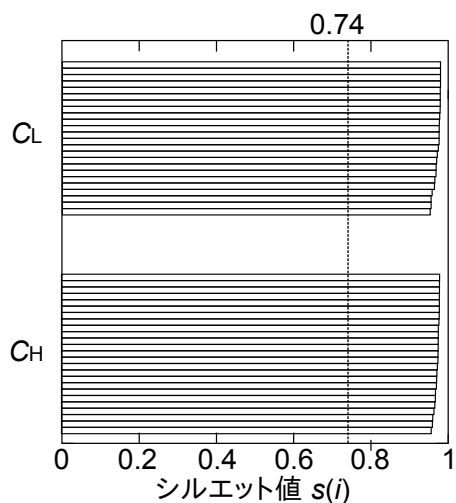


図 3.11: 図 3.10 におけるシルエットプロットの例

た場合を示している. 提案手法では, 1σ の時に歩留まり損失と故障見逃しがそれぞれ 79.93%, 0.35% である. 歩留まり損失が非常に大きく, IDDQ テストの判定基準を厳しく設定していることが分かる. 判定基準を緩めることで, 歩留まり損失を低減できる. 5σ の時に, 歩留まり損失を発生させず, 故障見逃しを 1.38% に抑制している. できている. NNR 法を用いた場合は, 4σ の時に歩留まり損失が発生しない. しかし, 故障見逃しが提案手法よりも 14 倍悪い数値となっている. NNR 法で良い結果が得られなかった理由として次の 2 点が考えられる.

表 3.2: 異なる良否判定基準に対する歩留まり損失と故障見逃し

		1σ	2σ	3σ	4σ	5σ	6σ	7σ	8σ	9σ
提案法	歩留まり損失 (%)	79.93	61.25	2.42	1.04	0.00	0.00	0.00	0.00	0.00
	故障見逃し (%)	0.35	0.35	1.04	1.04	1.38	1.73	1.73	2.77	2.77
NNR 法	歩留まり損失 (%)	9.00	5.88	2.77	0.00	0.00	0.00	0.00	0.00	0.00
	故障見逃し (%)	15.22	17.65	20.07	20.07	20.07	20.07	20.07	20.07	20.07

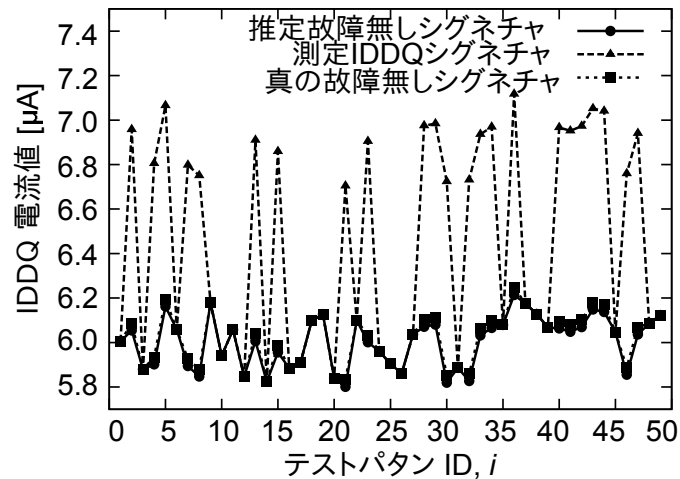


図 3.12: 故障無し IDDQ シグネチャの推定例

- 隣接チップ間のデバイスパラメータ値の差によるリーク電流値よりも小さなリーク故障サイズを持つ不良チップは検出できない
- 与えられたテストパターンで数パターンしかリーク故障が活性化されなかった場合、IDDQ シグネチャ内の故障リーク電流値が少ないため、そのチップの IDDQ 平均値が低くなり、良品チップの IDDQ 平均値との区別が難しい。

上記の理由によって、先端プロセスかつ大規模チップの場合は、NNR 法を用いて IDDQ リーク故障を検出することは難しいと考える。

図 3.12 に、故障無し IDDQ シグネチャの推定結果を示す。本例では、リーク故障サイズが $0.87\mu\text{A}$ でデバイスパラメータは $(3\text{mV}, 6\text{mV})$ である。この模擬チップは、クラスタリング方式フィルタにおいてはフェイル判定されていない。横軸はテストパターン ID で縦軸が IDDQ 電流値である。実線、三角形と破線、四角形と破線が、推定した故障無し IDDQ シグネチャ、測定 IDDQ シグネチャ、真の故障無し IDDQ シグネチャを表す。図から、推定した故障無し IDDQ シグネチャと真の故障無し IDDQ シグネチャが一致して

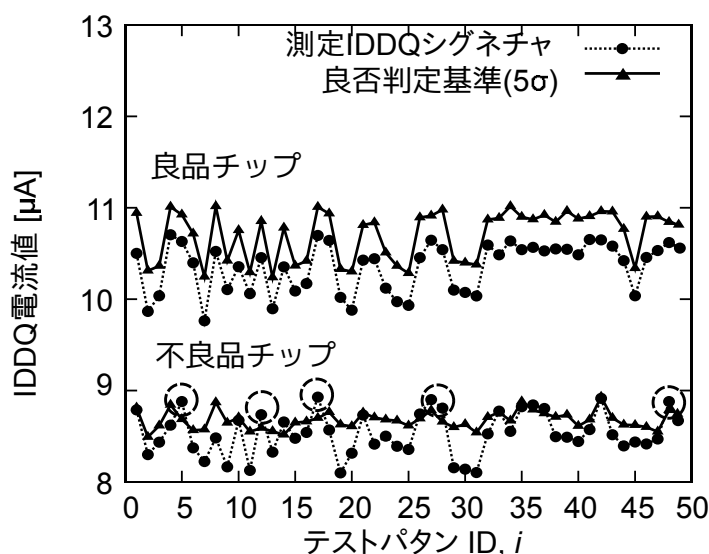


図 3.13: 2 模擬チップにおける IDDQ 良否判定基準の例．一方は良品チップで、もう一方は不良品チップである．

いることが分かる．本例では、提案手法の SA 法による故障無し IDDQ シグネチャ推定が正しく機能している．

図 3.13 に、良品チップと不良品チップの、測定 IDDQ シグネチャと良否判定基準を示す．図の下部の 2 曲線と上部の 2 曲線は、それぞれ良品チップと不良品チップを表す．両模擬チップともに、クラスタリング方式フィルタにおいてフェイル判定されていない．実線と破線はそれぞれ、 5σ と設定した場合のテスト良否判定基準値と IDDQ 電流シグネチャである．良品チップの例において、常に測定 IDDQ シグネチャはテストしきい値よりも低くなっており、本テストにおいてパスしたことを表す．一方、故障チップは、図 3.13 の円で示した点において、IDDQ シグネチャが IDDQ 電流しきい値を超えており、フェイルの判定となる．この不良チップでは、リーク故障サイズは $0.36\mu\text{A}$ で、 $(0\text{ mV}, 0\text{ mV})$ の時のチップの IDDQ シグネチャの平均電流値の 5% に相当し、非常に小さなリーク故障サイズである．不良品チップのデバイスパラメータは $(11\text{ mV}, 22\text{ mV})$ であり、デバイスパラメータ推定は、これを正しく推定できている．また、図 3.13 では、不良品チップよりも良品チップの方が IDDQ 電流値が大きい．このことから、提案手法はプロセスばらつきによってリーク電流値が大きい状況においても、不良リーク電流を判定することができる．このように、提案手法を用いることで、リーク電流値における、プロセスばらつきに起因する成分と故障による成分を分

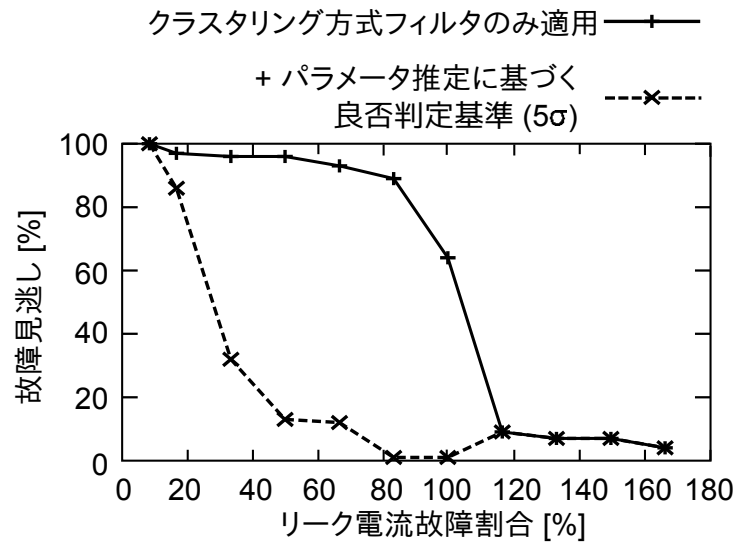


図 3.14: リーク故障サイズと故障見逃しの関係

離できる。

3.3.3 リーク故障サイズ毎の検出能力評価の結果

リーク故障サイズに対する提案手法の故障検出能力を示すため、図 3.14 に 100 不良チップに対する故障見逃し結果を示す。100 不良チップには単一縮退故障がランダムに埋め込まれている。全てのチップに対して、デバイスパラメータは (0 mV, 0 mV) でノミナル条件とした。挿入された故障のサイズは、0.5, 1.0, 2.0, ..., 10.0 μA とした。本実験における (0 mV, 0 mV), (-80 mV, 80 mV) の模擬 DUT の故障無し IDDQ シグネチャの平均 IDDQ 電流値はそれぞれ 6.02 μA , 46.52 μA である。よって、上記の 10.0 μA の故障は、特性ばらつきで増幅された IDDQ 電流値に比べても大きくない。図 3.14 において、横軸は、(0 mV, 0 mV) における平均電流 IDDQ 値に対するリーク故障サイズの割合で、ここではリーク電流故障割合として定義する。縦軸は、100 不良チップに対する故障見逃し率である。実線は、クラスタリング方式フィルタのみを適用した場合で、実線はさらに良否判定基準計算を適用した結果である。この時の判定基準は 5σ である。この図から、リーク電流故障割合が減少するとフェイル判定が難しくなっていることがわかる。また、リーク電流故障割合が 116 % 以上のリーク電流サイズの故障は、クラスタリング方式フィルタのみで 80 % 以上が検出できる。リーク電流故障割合が 40 % 以

上の場合だと、さらにテストしきい値計算を適用することで、80%以上のリーク電流サイズの故障が検出可能となる。リーク電流故障割合が20%未満の場合は、提案手法を適用したとしても検出が難しい。この場合においては、故障によるリーク電流増加とプロセスばらつきによるリーク電流増加を区別することが難しく、誤判定が生じている。

3.4 まとめ

本章では、第2章で提案したデバイスパラメータ推定を用いてデバイスパラメータを推定し、これに基づいて適応的に IDDQ テスト良否判定基準を決定する手法を提案した。IDDQ シグネチャを用いたデバイスパラメータ推定手法を用いてデバイスパラメータを DUT 毎に推定する。高精度に推定されたデバイスパラメータ値を用いて、DUT 毎に最適な良否判定基準を設定する。しかし、IDDQ テスト良否判定基準の計算のために、SA 法を採用しており、その計算時間が長いという課題がある。そこで、判定基準計算の前に、クラスタリング方式フィルタを適用し、大きなリーク故障のチップを事前に不良と判定する。これにより、良否判定計算が行われる DUT の数を削減し、全体の計算コストを削減する。本章では、以上の2段階の IDDQ テスト手法を提案した。計算機実験において歩留まり損失を発生させることなく故障見逃しを 1.38%に抑制し、従来提案されている NNR 法と比べて 14 倍のテスト品質を達成した。また、提案手法を適用することで、ノミナル条件時の IDDQ 電流の 40%の大きさの IDDQ 電流故障に対しては、その 80%以上を検出できる。

第4章

特性推定に基づく適応型パス遅延故障テスト

特性ばらつきに起因して発生するパラメトリック故障を低テストコストでテストするために、特性ばらつき推定に基づく適応型テストをパス遅延故障テストを提案する。パラメトリック故障の発生箇所は、特性ばらつきに応じて変動する。本章で提案する適応型テストは、特性パラメータ推定結果に応じて、パラメトリック故障が発生しやすいパスに対してパス遅延故障テストを行うため、テスト品質を保持しつつテストコストを削減できる。テストコストはテスト実行時間、すなわちテストパターン数に直結する。本章では、テストコストをテストパターン数で評価する。

4.1 はじめに

パラメトリック故障は、パスを構成する回路素子において、遅延ばらつきによる信号遅延が累積することで顕在化する。近年、微細プロセスにおける特性ばらつきの増大と高速化による設計マージンの削減により、パラメトリック不良が増加している [3-5]。ITRS ロードマップの 2011 年度版においても、プロセステクノロジーの進歩によりパラメトリック故障が増加することが警告されている [2]。パラメトリック故障をテストするために、パス遅延テストが適している [21,22]。パス遅延テストは、パス遅延故障モデルは、フリップフロップ間を結ぶパス上を対象としており、パス上の信号伝搬が蓄積する遅延をモデル化している。

文献 [41] では、製造されたチップ毎に特性ばらつきが異なるため、テストすべきパスを変えなければならないと主張している。ある特定の特性パラメータのみを考慮してテスト対象パスを選択した場合、パス遅延テストにおいては非効率になる。例えば、ワースト条件を考慮した場合、テストすべき

クリティカルパスは非常に多くなり、結果として、テストに用いられるテストパターン数も増加する。しかし、多くの DUT において、このテストパターン数は過剰である。

また、パス遅延テストを用いてパラメトリック故障をテストした時、テスト品質を表す有効な尺度が提案されていないことも大きな課題である。文献 [41, 70] では、PCM (Process space Coverage Metric) と TQM (Test Quality Metric) をパラメトリック故障検出能力を評価する尺度として提案している。しかし、これらの評価尺度は、総故障チップ数に対する検出可能な故障チップ数を計算した数値ではないため、ATPG ツールにおける故障検出率のように扱えない。よって、テスト品質を定量的に評価できない。

特性ばらつきや歩留まりによってテストを変更する手法として、適応型テストが提案されている [71–73]。文献 [71] では、ATE のテストプログラムを用いた実用的な適応型テストフローを提案している。文献 [72] では、テスト項目とテストフローを動的に変更する適応型テストが提案されている。文献 [73] では、ウェハテストに適用可能な適応型テストを提案している。しかし、これらの手法は、パラメトリック故障を考慮していない。

本章では、次の 2 つの新しい概念を含む適応型パス遅延故障テストを提案する。

- 適応型パス遅延テストでは、テスト前に DUT 毎にデバイスパラメータを推定し、その推定結果に基づきテストに用いるテストパターンを適応的に変更する。
- パラメトリック故障に対するテスト故障検出率を提案する。提案する尺度は、全パスにおけるタイミング違反確率とテストできたパスのタイミング違反確率の割合として定義する。本テスト尺度は、提案する適応型テストにおいても適用可能である。

従来のパス遅延故障テストでは、特定の特性パラメータのみを考慮した固定式パス遅延故障テストを行っている。計算機実験において、適応型パス遅延テストは、従来手法の固定式パス遅延故障テストと比べて、テスト品質を同等に保持しつつ、テストコストを 10 分の 1 以下に削減できることを示す。

以下、本章は次のように構成する。4.2 節で、提案する適応型パス遅延故障テストについて説明する。続いて、4.3 節にて、パラメトリック故障検出率を提案する。4.4 節にて、2M ゲート規模のベンチマーク回路と OpenCores ベンチマーク回路 [44] に対する適用結果を述べる。最後に、4.5 節にて本章をまとめる。

4.2 適応型パス遅延故障テストフロー

本節では，適応型パス遅延故障テストについて述べる．適応型パス遅延故障テストでは，デバイスパラメータに応じて最も良いテストを行うために，設計段階で，各特性パラメータ毎にテストパターンを生成する．続いて，DUT 毎にデバイスパラメータを推定する．テスト時に，推定したデバイスパラメータに応じてテストパターンを変更する．

この概念を実現する適応型テストフローを図 4.1 に示す．本フローは，設計フェーズとテストフェーズから構成される．

4.2.1 設計フェーズ

まず，設計フェーズにおいて，パスクラスタと呼ばれるテスト対象パスの集合を生成する．パスクラスタとは同じテスト対象パスを共有するパス群である．ATPG ツールを用いて，各パスクラスタに対して，テストパターンを生成する．パラメトリック故障に対するテストパタンのテスト品質を評価し，要求される品質を満足するならば，次のテストフェーズに移行する．

パスクラスタリングの目的は，テスト品質を保持しつつ，テストコストを削減することである．パスクラスタリングでは，デバイスパラメータのパス遅延感度を入力として用いる．パス遅延感度は，SSTA を用いて計算される．多くの SSTA ツールにおいて，パス遅延は次のような一次の線形式で表現される [18]．

$$d = \mu_d + \sum_{i=1}^n S_{p_i} \Delta p_i + rnd \quad (4.1)$$

ここで， d はパスの遅延値， μ_d は d の平均値， Δp_i は i 番目のデバイスパラメータ， S_{p_i} は Δp_i に対するパス遅延感度である． $N(0, \sigma_{rnd}^2)$ は局所ばらつきによるパス遅延の変動を表し，平均値が 0，標準偏差が σ_{rnd} の正規分布である．以下，提案する適応型テストでは，局所ばらつきを正規分布と仮定しているが，容易に任意の分布に拡張できる．

パスクラスタリングアルゴリズムとして，文献 [74,75] の手法が提案されている．本研究では，上藺らにより提案されたパスクラスタリングアルゴリズム [74] を用いることを想定している．本アルゴリズムでは，まず，パスクラスタリングの対象となるパスを抽出する．ここではテスト品質の低下を招くことがないように，テスト対象となり得るパスを悲観的に抽出する．このパス抽出の一例として， $\mu_{ss} + 3\sigma_{rnd}$ が対象回路のシステム周期を超えるパス

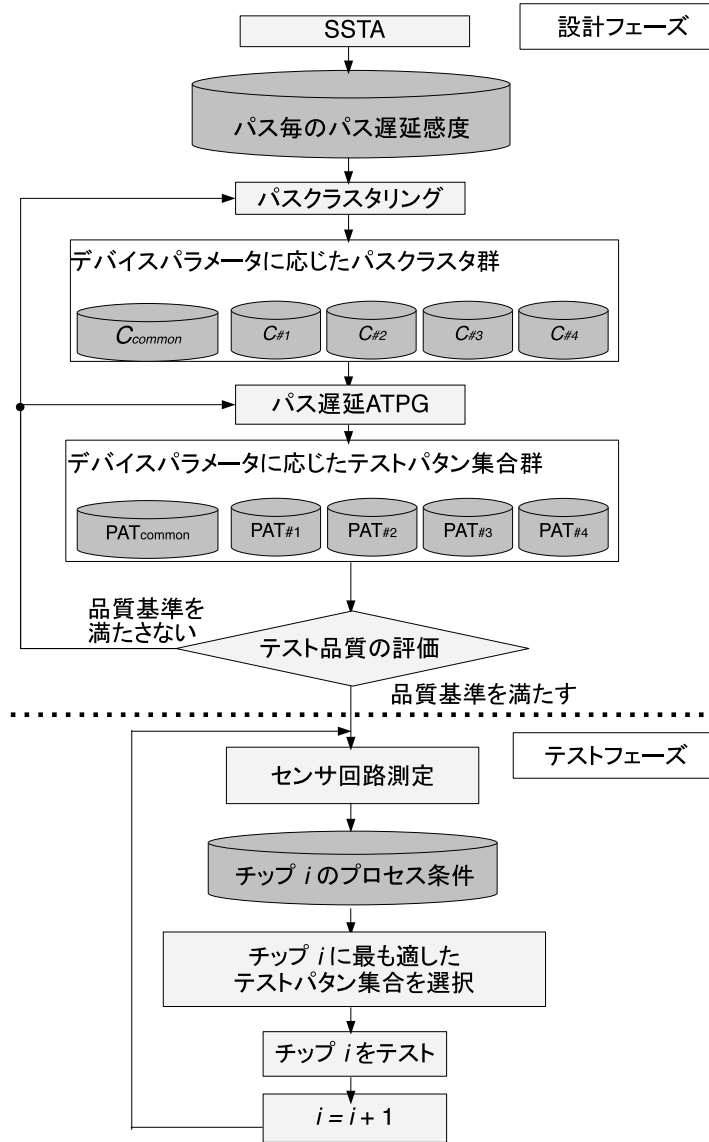


図 4.1: 適応型パス遅延テストフロー

を抽出する．ここで、 μ_{SS} はデバイスパラメータが SS コーナー条件の時のパス遅延値の平均である．パスクラスタリングは、これらのパス集合に対してまとめ上げを行う．パスクラスタリングでは、図 4.2 のように想定するデバイスパラメータ空間を小領域に分割する．図 4.2 では、 ΔV_{thp} と ΔV_{thn} の 2 次元空間で表現し、各小領域の刻み幅をそれぞれ d_m , d_n としている．この図において、ある小領域 c におけるテスト対象パス集合を K_c とする． K_c は、デバイスパラメータ ΔV_{thp} と ΔV_{thn} の組み合わせが小領域 c を指す時、

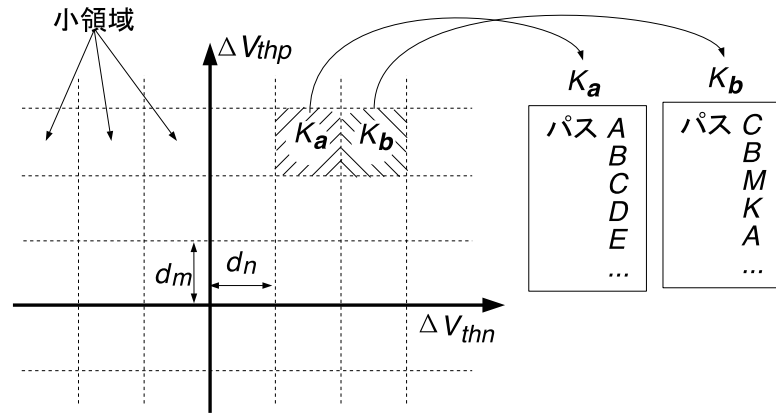


図 4.2: 各小領域におけるテスト対象パス集合 [74].

テスト対象となり得るパスの集合である。

小領域 a と b におけるパス集合 K_a と K_b をまとめ上げた場合を考える。推定デバイスパラメータ結果が小領域 a ，または b であった時のテスト対象パス集合は， $K_a \cup K_b$ である。推定デバイスパラメータが小領域 a であった時， $K_a \cup K_b - K_a$ は，テストする必要のない過剰なパス集合である。文献 [74] のパスクラスタリングアルゴリズムでは，この過剰なパス集合が小さくなるようにパスをまとめ上げる。以上の処理を，目標とするクラスタ数になるまで繰り返す。また，大域ばらつきを考慮した時，デバイスパラメータ空間の原点にある小領域が最も起こりやすい。パスクラスタリングアルゴリズムにおける距離では，過剰なパス数に小領域の生起確率で重み付けをし，確率的に過剰となるパス数が最小になるようにしている。

図 4.3 に，文献 [74] のアルゴリズムを適用した際のパスクラスタリング結果を示す。この時の d_x と d_y は共に 5mV である。 d_x と d_y はデバイスパラメータ推定手法の推定誤差により決定される。図 4.3 において，同じ記号は，同じパスクラスタに属することを示す。右下は SS コーナー条件を表し，左上は FF コーナー条件を表す。図 4.3 の凡例は，各パスクラスタにおけるテスト対象パス数を表す。一般的に，SS コーナーではテスト対象パス数が多く，FF コーナーではテスト対象パス数が少ない。図 4.3 においてもこの傾向が確認できる。FF コーナー条件でテスト対象となるパスは，他の多くのパスクラスタにおいてもテスト対象となる場合が多い。全クラスタでテスト対象となるパス群を共通パスクラスタとする。共通パスクラスタを用いることで，他クラスタは差分のみを考えればよい。ATE に格納するテストパターンを節約できる。図 4.1 では，共通パスクラスタを C_{common} とし， $C_{\#1}, \dots$

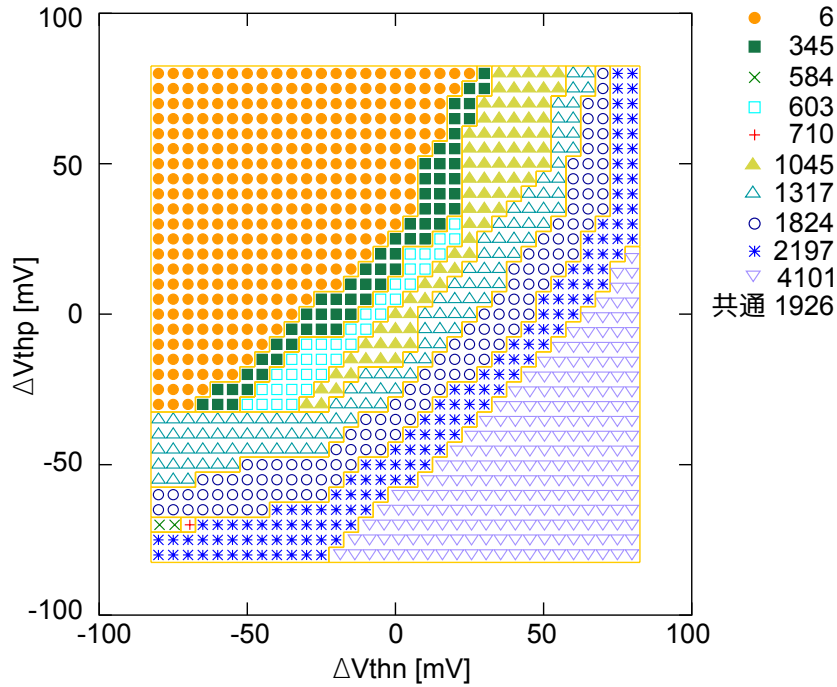


図 4.3: 文献 [74] におけるパスクラスタリング結果 (d_x , d_y は共に 5mV). 同じ記号は同じクラスタに属することを示す.

$C_{\#4}$ までのパスクラスタにまとめ上げている.

ここで, 図 4.3 におけるパスクラスタリングの効果について簡単に説明する. 共通パスクラスタ C_{common} のパス数は 1,926 である. デバイスパラメータ推定結果が SS 条件であった場合, テスト対象となるパス数は $1926 + 4101 = 6027$ である. ΔV_{thp} と ΔV_{thn} が, 平均 0mV で標準偏差 26.7mV の正規分布に従うと仮定すると, パラメータ空間におけるテスト対象パスの期待値は 2,984 パスである. 従来の単一条件のみを考慮したパス遅延テストで, SS コーナー条件をテスト条件した場合と比べると, 4,095 パスがテスト対象から削減できたことになる.

パスクラスタリングの後, ATPG ツールを用いてパスクラスタ毎にテストパターンを生成する. 図 4.1 では, $\text{PAT}_{\text{common}}, \text{PAT}_{\#1}, \dots, \text{PAT}_{\#4}$ の 5 テストパターン集合を得ている. $\text{PAT}_{\text{common}}$ は共通パスクラスタ C_{common} に対する共通テストパターン集合で, $\text{PAT}_{\#1}, \dots, \text{PAT}_{\#4}$ は $C_{\#1}, \dots, C_{\#4}$ に対するテストパターンである.

テストパターン生成後, テストパターンのパラメトリック故障に対する検出能力, すなわちパラメトリック故障検出率を用いて, テスト品質を評価する.

パラメトリック故障検出率については、4.3 節にて詳細に述べる。ここで、パラメトリック故障検出率が要求される品質を満たしていれば、続くテストフェーズへ移行し、満たさない場合は、パスクラスタリング、あるいはパス遅延 ATPG へ戻る。一般的に、回路内部には、機能的に活性化しない機能的活性不可能パスが多く存在することが知られている [76]。このようなパスは、本来、フォルスパスとしてテスト対象から除外しなければならない。パラメトリック故障検出率を向上させるために、設計段階でのフォルスパス判定手法の適用が有効である [77]。

4.2.2 テストフェーズ

テストフェーズでは、まず、チップ毎にデバイスパラメータ p_i を推定する。その後、パラメータ値に応じて、使用するテストパターンを選択する（共通テストパターン PAT_{common} とその他のテストパターンを 1 つ選択する）。図 4.3 から、SS コーナー条件で対象となるテスト対象パスは他の条件でテスト対象となるパスを包含する。よって、SS コーナー条件を考慮したテストパタンのみを ATE メモリに格納するだけで、他パスクラスタのテストにも対応できる。テストでは、図 4.4 に示すように、センサ回路から得られた推定デバイスパラメータ値を用いてテストパターンを選択する。共通パスクラスタに対応するテストパターン PAT_{common} は常に使用される。パスクラスタリングの結果はルックアップテーブルとして、テストパターンと同様に ATE に格納される。例えば、図 4.1 において、デバイスパラメータがパスクラスタ $C_{\#3}$ に対応する値であった場合、テストパターン $PAT_{\#3}$ と PAT_{common} が使用される。最新の ATE はアプリケーションプログラミングインターフェース (API) 機能を備えており、このようなテストパタンの変更はテスト用の制御プログラミングを実装することで実現できる [78]。ATE メモリに格納されるテストパターンは、SS コーナー条件のテストパタンのみである。適応型パス遅延故障テストとして追加でメモリに格納される情報は、パスクラスタリング結果のルックアップテーブルとテストパターン選択を制御する命令文のみである。

4.3 パラメトリック故障検出率

テストパターン生成後、そのテスト品質を定量的に評価する必要がある。本節では、パラメトリック故障に対するテスト品質評価尺度であるパラメトリック故障検出率を提案する。提案する検出率は、デバイスパラメータに応

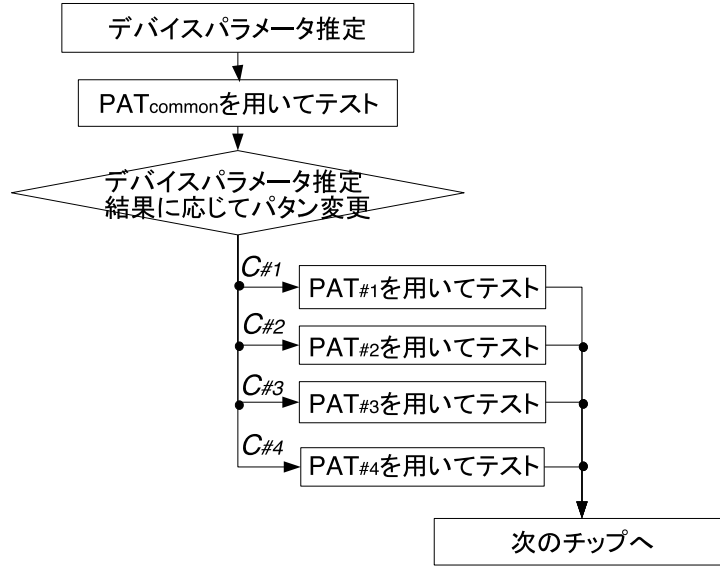


図 4.4: テストフローチャート

じてテストパターンを変更する適応型テストフローにおいても対応している。

提案するパラメトリック故障検出率の基本的な考え方を 4.3.1 節で述べ、4.3.2 節で適応型テストへの拡張について述べる。

4.3.1 パラメトリック故障検出率の基本概念

提案するパラメトリック故障検出率は、文献 [37,41] で提案されているパス遅延故障検出率のように、パス遅延分布の確率密度の概念を導入する。パス遅延値は、式 (4.1) で計算される。テスト対象回路における j 番目のパス P_j のパス遅延値を D_{P_j} とする。ここで、 $1 < j < N$ とする。 N は回路における全パス数である。 T_{sys} はテスト対象回路のシステムクロック周期とする。パス P_j にパラメトリック故障が発生しない確率は次のようになる。

$$P\{D_{P_j} \leq T_{\text{sys}}\} \quad (4.2)$$

よって、全パスにおいてパラメトリック故障が発生しない確率は、次のようになる。

$$P\{D_{P_1} \leq T_{\text{sys}}, D_{P_2} \leq T_{\text{sys}}, \dots, \text{and } D_{P_N} \leq T_{\text{sys}}\} \quad (4.3)$$

ここで、 $P\{D_{P_1} \leq T_{\text{sys}}, D_{P_2} \leq T_{\text{sys}}, \dots, \text{and } D_{P_N} \leq T_{\text{sys}}\}$ は $P\{D_{P_1} \leq T_{\text{sys}}\}, P\{D_{P_2} \leq T_{\text{sys}}\}, \dots, P\{D_{P_N} \leq T_{\text{sys}}\}$ の同時確率である。SSTA 内の計算で行われているように、同時確率を統計的 MAX 演算を用いて近似すると、式 (4.3) は次のようになる [18]。

$$P\{\max_j(D_{P_j}) \leq T_{\text{sys}}\} \quad (4.4)$$

よって、パラメトリック故障の発生確率 pf_{all} は、式 (4.5) のようになる。

$$\begin{aligned} pf_{\text{all}} &= 1 - P\{\max_j(D_{P_j}) \leq T_{\text{sys}}\} \\ &= P\{\max_j(D_{P_j}) > T_{\text{sys}}\} \end{aligned} \quad (4.5)$$

一方で、テストパタン S_T が与えられ、 S_T でテストされるパス P_T におけるパラメトリック故障の発生確率 pf_{tested} は次のように計算できる。

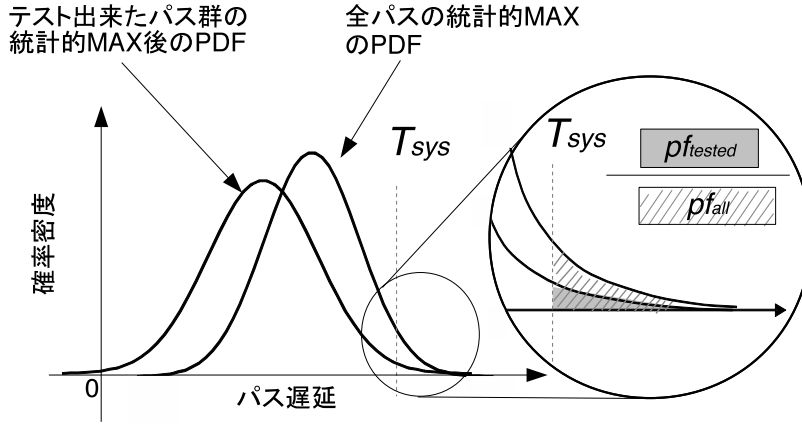
$$pf_{\text{tested}} = P\{\max_{P_j \in P_T}(D_{P_j}) > T_{\text{sys}}\} \quad (4.6)$$

ここで、テストパタン S_T におけるパラメトリック故障検出率 pf_c を次のように定義する。

$$\begin{aligned} pf_c &= \frac{P\{\max_{P_j \in P_T}(D_{P_j}) > T_{\text{sys}}\}}{P\{\max_j(D_{P_j}) > T_{\text{sys}}\}} \\ &= \frac{pf_{\text{tested}}}{pf_{\text{all}}} \end{aligned} \quad (4.7)$$

pf_{all} 及び pf_{tested} は SSTA ツールを用いて計算可能である。 P_T は、テストされたパスであるため、式 (4.7) の分子は P_T に依存する。図 4.5 に pf_c の概念図を示す。 pf_c は、全パスにおけるパラメトリック故障率とテストしたパスにおけるパラメトリック故障率の割合として定義される。

高い pf_c を得るために、小スラック、かつ分散の大きいパスをテストする必要がある。例えば、 $\frac{(T_{\text{sys}} - \mu_d)}{\sigma_{\text{rnd}}}$ が小さいパスをテストすべきである。この数値が小さければ、そのパスが T_{sys} を超えてタイミング違反となる確率が高い。また、 pf_c はパスの遅延値とばらつきを考慮しているため、パス遅延故障検出率が低くても、小スラックパスがテストされていれば pf_c は高い値となり得る。

図 4.5: pfc の概念図

4.3.2 適応型テストへの拡張

pfc は P_T を固定した場合、つまり単一のテストパタンのみを用いた場合のパラメトリック故障検出率である。適応型テストでは、複数のテストパタンを DUT 毎に変更するため、 pfc を拡張する必要がある。

適応型テストフローでは、テスト前にデバイスパラメータ値を推定する。よって、デバイスパラメータ値によって、 D_{P_j} が変化し、 pf_{all} と pf_{tested} もまた変化する。また、テストパタンを変更することで、 P_T も異なる。デバイスパラメータ推定後の D_{P_j} , pf_{all} , pf_{tested} , pfc を D'_{P_j} , pf'_{all} , pf'_{tested} , pfc' とする。 pfc' は式 (4.8) のようになる。

$$\begin{aligned}
 pfc' &= \frac{P\{\max_{P_j \in S_T}(D'_{P_j}) > T_{sys}\}}{P\{\max_j(D'_{P_j}) > T_{sys}\}} \\
 &= \frac{pf'_{tested}}{pf'_{all}} \quad (4.8)
 \end{aligned}$$

式 (4.8) において、デバイスパラメータが推定されているため、 pfc' は pfc より高精度になる。

ここで、パラメトリック故障率をデバイスパラメータ空間における期待値として考える。Prob(x, y) を $\Delta V_{thp} = x$ と $\Delta V_{thn} = y$ となる生起確率とする。 $pf'_{all}(x, y)$ と $pf'_{tested}(x, y)$ を $(\Delta V_{thp}, \Delta V_{thn}) = (x, y)$ の時の pf'_{all} と pf'_{tested} とする。これにより、デバイスパラメータ空間における pfc の期待値 PFC

は次のように与えられる.

$$PFC = \frac{\int_{all\ x} \int_{all\ y} pf'_{tested}(x, y) \cdot \text{Prob}(x, y) dydx}{\int_{all\ x} \int_{all\ y} pf'_{all}(x, y) \cdot \text{Prob}(x, y) dydx} \quad (4.9)$$

しかし、式 (4.9) は連続関数の積分値として表現されているため、計算機上での実装が難しい．そこで、式 (4.9) を離散値として近似する．図 4.2 のように、 ΔV_{thp} と ΔV_{thn} を幅 d_m , d_n で分割する．この時の分割数を m 個, n 個とする、これにより、 PFC は次のように近似される．

$$PFC = \frac{\sum_i^m \sum_j^n pf'_{tested}(i, j) \cdot \text{Prob}(i, j) \cdot d_m d_n}{\sum_i^m \sum_j^n pf'_{all}(i, j) \cdot \text{Prob}(i, j) \cdot d_m d_n} \quad (4.10)$$

pf' は特定のデバイスパラメータ値のみを考慮したテスト品質尺度であるのに対し、 PFC は想定するデバイスパラメータ空間全てを考慮している．提案する適応型テストフローでは、 PFC を用いてテストパタンの品質を評価する．この例では、 ΔV_{thp} と ΔV_{thn} を用いて説明したが、別のパラメータも同様に扱うことができる．

4.4 シミュレーション実験

本節では次の2実験を行い、特性の特性パラメータ値のみを考慮した従来のパス遅延故障テストと比較して、提案する適応型テストではテスト品質を保持しつつ、テストコストを削減できることを示す．

理想クラスタ時の適応型テスト:

チップ毎に最適なテストパターンが適用される理想的な適応型テスト

パスクラスタリング [74] を用いた適応型テスト:

パスクラスタリング適用によりパスクラスタを決定する適応型テスト

4.4.1 理想クラスタ時の適応型テスト

実験準備

本節では、2M ゲート規模のベンチマーク回路 STARC03 に適応型テストを適用した結果を述べる．STARC03 の仕様を表 4.1 に示す．従来のパス遅

表 4.1: STARC03 仕様 [25].

parameter	value
ゲート数	2 M (2 入力 NAND 換算)
フリップフロップ数	69,180
クロックドメイン数	4
システムクロック周波数/システムクロック周期	
CLK_A	250 MHz/4 ns
CLK_A1	125 MHz/8 ns
CLK_A2	62.5 MHz/16 ns
CLK_B	28 MHz/36 ns

延故障テストでは，特定の特性ばらつきパラメータ値のみを考慮している．本実験では，従来のパス遅延テスト（conv-nom, conv-worst）と適応型テスト（ours）の比較を行う．

conv-nom: ノミナル 条件においてワースト 50,000 パスを抽出してパス遅延テストパターン生成を行う．DUT のデバイスパラメータ値によらず常にこのテストパターンを使用する．

conv-worst: SS 条件においてワースト 50,000 パスを抽出してパス遅延テストパターン生成を行う．DUT のデバイスパラメータによらず常に本テストパターンを使用する．

ours: 全小領域毎にワースト 50,000 パスを抽出してパス遅延テストパターン生成を行う．デバイスパラメータに応じてテストパターンを変更する．

これら 3 種のテスト手法において，常に 50,000 パスがテスト対象となる．また，本実験の ours ではパスクラスタリングを適用しておらず，小領域の数だけパスクラスタが存在することを想定している．本実験における適応型テスト手法では，常に最適なテストパターンが適用されるため，理想的な適応型テストといえる．

本実験では，90-nm の標準セルセットを用いて論理合成された STARC03 を用いた．配線遅延は，配線に接続されるセル数とセルの駆動能力から見積もられる配線遅延モデルを考慮し，固定として扱った．90-nm 標準セルセットでは，nMOS と pMOS のしきい値電圧がばらつく想定し，局所ばらつきの大きさは文献 [55] に従って設定した．デバイスパラメータ ΔV_{thp} と ΔV_{thn} は平均値 0 mV，標準偏差 26.6 mV の正規分布に従ってばらつく想定

表 4.2: 従来手法 (conv-nom, conv-worst) におけるパス遅延 ATPG 結果

	conv-nom	conv-worst
テストパターン数	215	200
故障検出率	31.48 %	31.24 %

表 4.3: 適応型テスト (ours) におけるパス遅延 ATPG 結果

		ΔV_{thn} (mV)				
		-80	-40	0	40	80
ΔV_{thp} (mV)	80	46.52% 174	45.74% 177	46.62% 196	44.74% 198	43.44% 190
	40	46.32% 161	46.32% 177	46.50% 181	46.38% 200	44.86% 196
	0	47.38% 162	46.52% 171	46.08% 187	46.78% 195	46.68% 202
	-40	48.22% 149	47.94% 165	46.88% 171	47.18% 170	46.94% 179
	-80	47.32% 116	46.98% 117	46.98% 126	47.84% 156	48.16% 159

している．ここで、 ΔV_{thp} と ΔV_{thn} の間の相関は考慮していない．STARC03 の CLK_A 系をパス遅延テストの対象とした． CLK_A の T_{sys} は 3.657 ns とし、これは 3σ 設計が適用されたことを想定している．すなわち、STARC03 のパラメトリック故障発生確率は 1,350 PPM である． CLK_A 系のパス数は 436,613 である．

従来手法と適応型テストにおけるテスト対象パスの抽出は、それぞれ、商用の STA ツールと SSTA ツールを用いた [58,79]．デバイスパラメータ空間では、次の ΔV_{thp} と ΔV_{thn} の全組合せ 25 個とした．

$$\Delta V_{\text{thn}} = (-80 \text{ mV}, -40 \text{ mV}, 0 \text{ mV}, 40 \text{ mV}, 80 \text{ mV})$$

$$\Delta V_{\text{thp}} = (-80 \text{ mV}, -40 \text{ mV}, 0 \text{ mV}, 40 \text{ mV}, 80 \text{ mV})$$

表 4.4: 従来手法と提案手法の PFC 比較

	conv-nom	conv-worst	ours
PFC	17.67 %	33.47 %	73.57 %

表 4.5: ΔV_{thp} と ΔV_{thn} の同時確率

		ΔV_{thn} (mV)				
		-80	-40	0	40	80
ΔV_{thp} (mV)	-80	4.41e-05	1.29e-03	3.98e-03	1.29e-03	4.41e-05
	-40	1.29e-03	3.77e-02	1.16e-01	3.77e-02	1.29e-03
	0	3.98e-03	1.16e-01	3.58e-01	1.16e-01	3.98e-03
	40	1.29e-03	3.77e-02	1.16e-01	3.77e-02	1.29e-03
	80	4.41e-05	1.29e-03	3.98e-03	1.29e-03	4.41e-05

pfc' と PFC の計算は, SSTA ツールの統計的 MAX 演算による歩留まり計算機能を用いた. 適応型パス遅延テストのテスト対象パスの抽出において, $\frac{(T_{sys}-\mu)}{\sigma_{rnd}}$ の値が小さいパスの順に 50,000 パスを抽出した.

商用 ATPG ツールを用いてパス遅延故障テストパターンを生成した [53]. パラメトリック故障をテストするために, パス上の微小な遅延変動を確実に終点フリップフロップで観測する必要がある. そこで, 本実験では, ロバスト検出条件を使用した [57]. 従来手法におけるパス遅延故障 ATPG の結果を表 4.2 に示す. パス遅延故障検出率は, テスト対象パス数 (50,000) と検出可能パス数の割合と定義している. 表 4.3 に, 適応型テストフローにおける小領域毎のパス遅延故障 ATPG 結果を示す. 表の各格子の上部にパス遅延検出率を示し, 下部にテストパターン数を示す. 表 4.3 において, パス遅延故障検出率の平均値は 46.57 % で, テストパターン数の平均値は 171 である.

理想クラスタ時の適応型テストの適用結果

表 4.4 に, 従来手法と適応型パス遅延テストの PFC 値を示す. ΔV_{thp} と ΔV_{thn} は正規分布に従ってばらつくと仮定しており, 各小領域の同時確率を計算すると表 4.5 のようになる. 例えば, $(\Delta V_{thp}, \Delta V_{thn}) = (80 \text{ mV}, -80 \text{ mV})$

表 4.6: 従来手法 conv-nom における小領域毎の pfc'

		$\Delta V_{\text{thn}} \text{ (mV)}$				
		-80	-40	0	40	80
$\Delta V_{\text{thp}} \text{ (mV)}$	-80	N/A	N/A	N/A	N/A	N/A
	-40	N/A	N/A	N/A	N/A	0 %
	0	N/A	N/A	N/A	0 %	0 %
	40	N/A	N/A	0 %	0 %	10 %
	80	N/A	0 %	0 %	26 %	100 %

表 4.7: 従来手法 conv-worst における小領域毎の pfc'

		$\Delta V_{\text{thn}} \text{ (mV)}$				
		-80	-40	0	40	80
$\Delta V_{\text{thp}} \text{ (mV)}$	-80	N/A	N/A	N/A	N/A	N/A
	-40	N/A	N/A	N/A	N/A	0 %
	0	N/A	N/A	N/A	0 %	0 %
	40	N/A	N/A	0 %	0 %	12 %
	80	N/A	0 %	0 %	57 %	100 %

の発生確率 $\text{Prob}(80, -80) = 4.41e - 05$ になる．小領域毎の発生確率を考慮し，式 (4.10) を用いて従来手法と適応型パス遅延テストの PFC を計算した．表 4.4 から，提案手法の PFC は，conv-nom，conv-worst と比べて 4 倍，2 倍となっている．

続いて，表 4.6–4.8 に，小領域毎の pfc' を示す．“N/A” は pfc'_{all} がほぼゼロであるため計算できないことを表す．これらの表から，全ての小領域において，提案手法 ours は従来手法と比べて同等以上の pfc' が得られていることが分かる．この結果から，適応型テストを用いることで，従来よりも高いテスト品質が得られている．

図 4.6 に，小領域 $(\Delta V_{\text{thp}}, \Delta V_{\text{thn}}) = (-40 \text{ mV}, 80 \text{ mV})$ におけるテストパターンに対する pfc' の関係を示す．同様の傾向が他の小領域においても見られる．横軸がテストパターン数で縦軸が pfc' である．実線が適応型テスト手法を表し，太い破線と細い破線がそれぞれ従来手法 conv-nom と conv-worst

表 4.8: 提案手法 ours における小領域毎の pfc'

		$\Delta V_{\text{thn}} \text{ (mV)}$				
		-80	-40	0	40	80
ΔV_{thp} (mV)	-80	N/A	N/A	N/A	N/A	N/A
	-40	N/A	N/A	N/A	N/A	0%
	0	N/A	N/A	N/A	0%	0%
	40	N/A	N/A	0%	1%	54%
	80	N/A	0%	39%	100%	100%

を表す。図 4.6 から、適応型テスト手法は、テストパターンに対して急速に上昇し、常に従来手法と比べて pfc' が高い。これは、提案手法を用いることで、少ないテストパターン数で高いテスト品質が得られることを示している。本実験では、4.2.1 節で述べたパスクラスタリングは適用されていない。パスクラスタリングを適用することで、 pfc' が飽和しているテストパターンは除外され、テスト品質を保持しつつテストコストが削減されることが期待される。

4.4.2 パスクラスタリングを用いた適応型テスト

パスクラスタリングを適用した提案手法の効果を確認するため、OpenCores ベンチマーク回路 [44] を用いたシミュレーション実験を行った。本節では、商用の 65-nm プロセスの標準セルライブラリを用いて設計した Ethernet 回路に対する適用結果を示す。パスクラスタリングを適用することで、従来手法と比べて、テスト品質を保持しつつテストコストを削減できることを示す。

実験準備

本実験では、Ethernet 回路を 65-nm の標準セルセットを用いて論理合成した [80]。配線遅延は、配線に接続されるセル数とセルの駆動能力から見積もられる配線遅延モデルを考慮し、固定として扱った。65-nm 標準セルセットでは、nMOS と pMOS のしきい値電圧がばらつく想定し、デバイスパラメータ ΔV_{thp} と ΔV_{thn} は平均値 0 mV、標準偏差 26.6 mV の正規分布

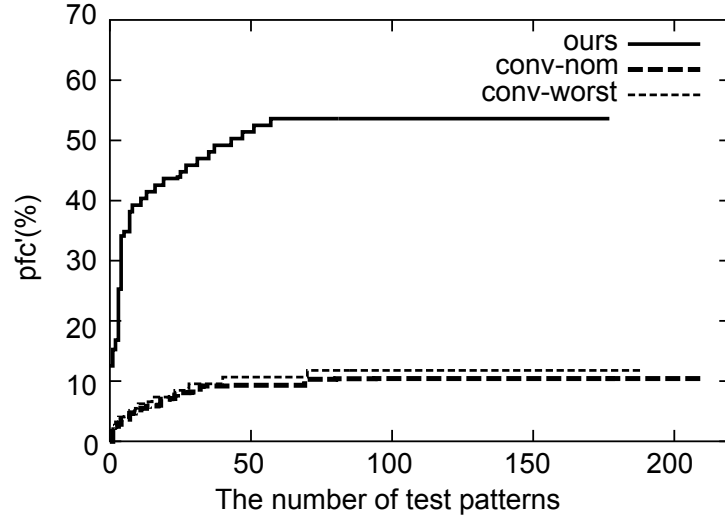


図 4.6: $(\Delta V_{\text{thn}}, \Delta V_{\text{thp}}) = (80 \text{ mV}, -40 \text{ mV})$ の時の、テストパターン数に対する pfc' の関係

に従ってばらつく想定している。文献 [59] において、65-nm プロセスの局所ばらつき成分は大域ばらつき成分の 60% の大きさであることが報告されている。局所ばらつき成分はこれに従って設定した。Ethernet 回路は送信側と受信側のクロック系統がある。本実験では、受信側のクロック系統 `mr_x_clk_pad_i` のパスを対象とし、`mr_x_clk_pad_i` の T_{sys} は 1.58 ns としている。これは実験 1 同様、 3σ 設計が適用されたことを想定している。

パス遅延感度を計算するために SSTA ツールを用いる代わりに、SPICE による回路シミュレーションを行った [54]。市販 ATPG ツールを用いて、パス遅延テストパターンを生成した [53]。実験 1 と同様に、パス遅延検出条件はロバスト条件とした。パスクラスタリング法は C 言語で実装した。文献 [74] は貪欲法を用いているが、クラスタリング時間を短縮するために、本実験で実装したパスクラスタリングは k-means++ 法 [81] を用いた。パスクラスタリングにおいて、デバイスパラメータ空間を ΔV_{thp} と ΔV_{thn} の 2 次元で考え、それぞれの範囲を -80 mV から $+80 \text{ mV}$ とする。第 2 章における推定結果から、 $d_m = d_n = 5 \text{ mV}$ と想定した。 pfc' を計算する際の統計的 MAX 演算は Monte Carlo シミュレーションで代用し、その試行回数は 1,000,000 回とした。

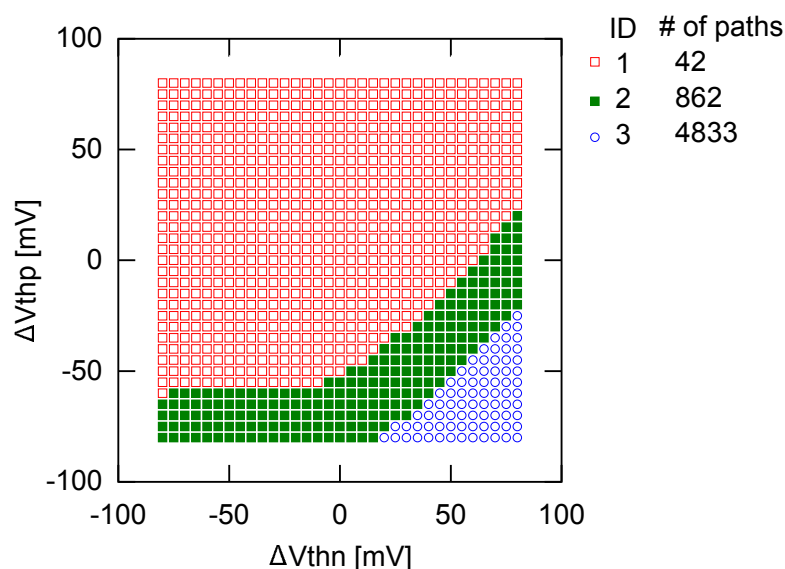


図 4.7: クラスタ数3の時のパスクラスタリング結果

パスクラスタリング [74] を用いた適応型テストの適用結果

本実験では、パスクラスタリングアルゴリズムで設定するクラスタ数を1, 3, 5, 10の4種類とした。クラスタ数が1の時は、従来手法におけるconv-worstと等価である。まず、図4.7から図4.9にクラスタ数が3, 5, 10の時のパスクラスタリングの適用結果を示す。横軸と縦軸はそれぞれ ΔV_{thn} と ΔV_{thp} を表し、同じ記号の小領域は同じパスクラスタに属することを示している。また、凡例において各パスクラスタのID番号とそのクラスタにおけるパス数を示している。各図において、右下と左上がそれぞれSSコーナーをFFコーナー表しており、図4.3と同様に、FFコーナーとSSコーナーを結ぶ線上でパス数が増加していることが分かる。クラスタ数が増える毎に、SSコーナー付近で細かくパスクラスタが形成されている。いずれのクラスタ数においてもSSコーナーでは4,833パスがテスト対象パスとなっている。クラスタ数が1の時は、全小領域においてこの4,833パスがテスト対象となる。

続いて、各パスクラスタに対してパス遅延故障 ATPG を行った結果を示す。図4.10(a)と4.10(b)にテストパターン数と故障検出率を示す。両図の横軸はパスクラスタのIDを表し、各記号はクラスタ数を1, 3, 5, 10とした時に対応している。例えば、クラスタ数を3とした場合はクラスタIDは1から3までの結果を示している。図4.10(a)と4.10(b)の縦軸は、それぞれテ

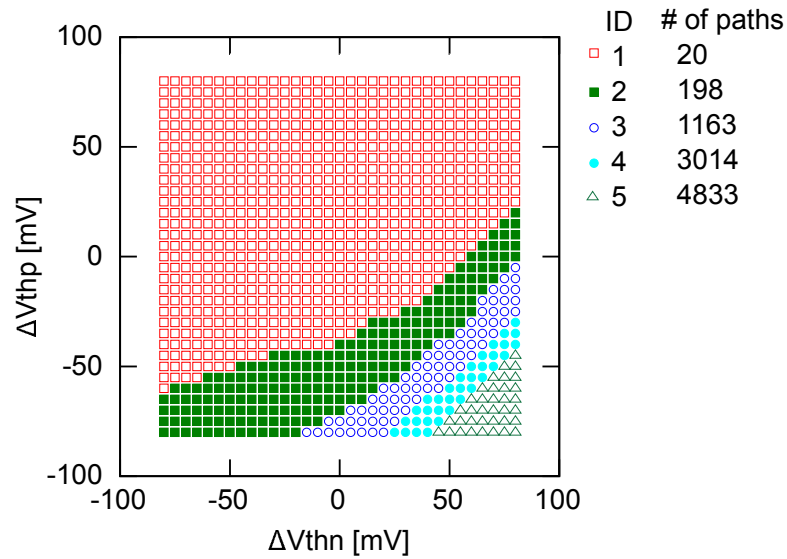


図 4.8: クラスタ数 5 の時のパスクラスタリング結果

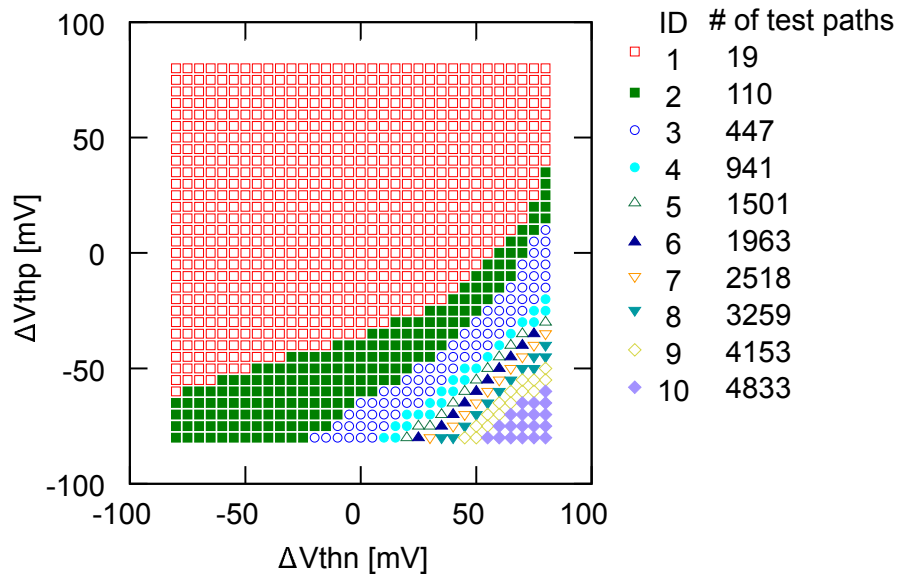
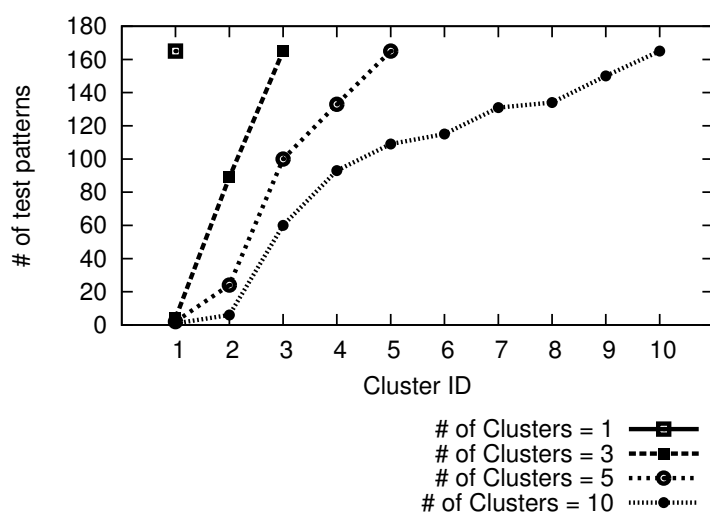
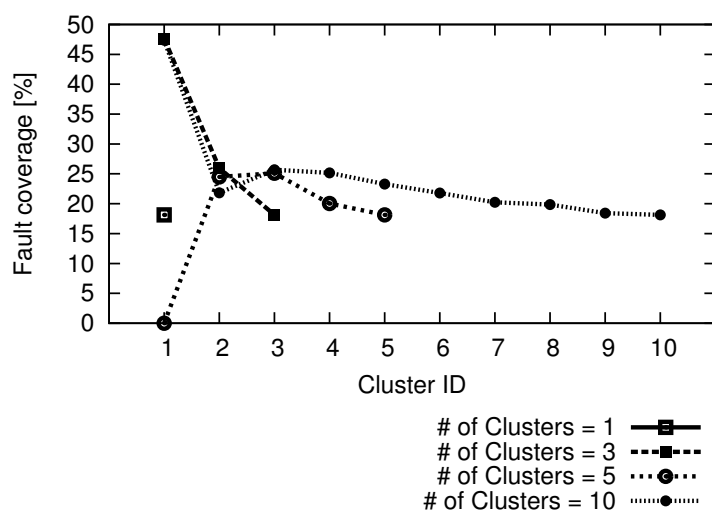


図 4.9: クラスタ数 10 の時のパスクラスタリング結果

ストパターン数と故障検出率である。SS コーナーにあるパスクラスタ ID の時は、いずれも 4,833 パスが対象となるため、これをテストするテストパタンの数はいずれも 164 である。パス遅延故障検出率の定義は、実験 1 同様、テスト対象となったパス数とロバスト検出条件で検出できたパス数である。クラスタ ID が増えることで、テスト対象パス数は単調増加しているため、



(a) パス遅延故障テストパターン数



(b) パス遅延故障検出率

図 4.10: 各パスクラスタに対するパス遅延テストパターン数とパス遅延故障検出率

図 4.10(a) においては, 単調に増加している. しかし, 図 4.10(b) では, この傾向は見られず, テスト対象パス数が増加することで, 故障検出率 20% に収束している.

最後に, クラスタ数を変化させた時の適応型テストフローにおける *PFC* と

表 4.9: パスクラスタリングを用いた適応型テスト結果

クラスタ数	1 (従来)	3	5	10
$PFC(\%)$	83.60	83.60	83.60	83.60
テストパターン数期待値	164	9.41	5.54	3.07
パターン数期待値従来比 (%)	100	5.73	3.38	1.87

テストパターン数の期待値を示す。PFC 計算においては、表 4.5 同様に、 ΔV_{thp} と ΔV_{thn} の同時確率を計算した。テストパタンの期待値は、各小領域の生起確率が分かっていることから、全小領域において適用されるテストパターン数の期待値を計算した。表 4.9 に、クラスタ数を 1, 3, 5, 10 とした時の PFC とテストパターン数期待値を示す。表から、いずれのクラスタ数においても $PFC = 83.60\%$ と高い数値を示しており、クラスタ数によるテスト品質の劣化は見られない。続いて、テストパターン数期待値について述べる。クラスタ数を 1 とした時は従来手法に対応し、テストパターン数の期待値は 164 である。これに対して、クラスタ数を 3, 5, 10 と増加させることで、テストパタンの期待値は急激に削減できている。クラスタ数が 10 の時に、従来手法との比で 1.87% にまで削減できている。クラスタ数を 3 の時においても、5.73% と 10 分の 1 以下となっており、提案手法によるテストコスト削減効果が高いことが分かる。以上の結果から、パスクラスタリングを用いた適応型テスト手法を適用することで、テスト品質を従来手法と同等に保ちつつ、テストパターン数期待値を削減できる。

4.5 まとめ

本章では、デバイスパラメータ推定に基づく適応的パス遅延故障テストを提案した。提案する適応型テストフローにおいては、テストの前に DUT 毎のデバイスパラメータ値を推定する。推定したデバイスパラメータに応じて、テストに用いるテストパターンを適応的に変更する。さらに、本章では、パラメトリック故障検出率を提案した。この検出率は、テスト対象回路の全パスにおけるパラメトリック故障確率とテストされたパスにおけるパラメトリック故障発生確率の比として定義される。計算機実験において、SS 条件のみを考慮した従来のパス遅延故障テストと比べて、テスト品質を同等に保持しつつ、テストコスト（テストパターン数の期待値）を 10 分の 1 以下に削減できることを示した。

第5章

結論

5.1 研究成果のまとめ

LSI 製造技術の進歩により LSI の大規模化，高性能化が進んでいる一方で，トランジスタのしきい値電圧やドレイン電流などの特性がばらつく特性ばらつきが問題となっている．特に，LSI テストにおいては，特性ばらつきの影響で製品仕様を満たさなくなるパラメトリック故障の増大，テストにおける良否判定基準の設定が困難になることが課題となっている．本論文では，LSI テストにおける，テスト品質の向上とテストに要するコスト削減を目的として，特性パラメータ推定に基づく適応型テスト手法を提案し，従来からある IDDQ テストとパス遅延テストに適用した．適応型テストを適用することで，既存の固定的なテスト手法と比べてテスト品質の向上とテストコストの削減が可能となった．

第2章では，適応型テストのための特性パラメータ推定手法を提案した．本章では，2種類の推定手法を提案した．1つは，量産テストにおいて広く適用されている IDDQ テストの測定時に得られる IDDQ シグネチャを用いた手法で，もう1つは， F_{\max} 値を用いる F_{\max} テストの測定結果を用いた手法である．これらの手法において，推定に必要な情報は全てテストから得られるため，リングオシレータなどの測定用回路は不要となる．IDDQ シグネチャを用いた推定手法では，事前に推定対象回路の統計的リーク電流分布を取得し，この情報と IDDQ テスト時に測定した IDDQ シグネチャをベイズ推定に適用して，特性パラメータを推定した． F_{\max} テストの情報を用いた手法では，IDDQ シグネチャを用いた手法と同様に，統計的な最大動作周波数の分布を取得した． F_{\max} テストより得られた測定 F_{\max} 値から，ベイズ推定，最尤推定法を用いて特性パラメータを推定した．これら2手法は同時に適用できるため，推定結果を検証しながら用いることができ，特性推定の高信頼化に繋がる．両手法とも，トランジスタのしきい値電圧を誤差 5mV 以

内で推定できた。

第3章では、特性パラメータ推定結果に基づく IDDQ テストしきい値の決定手法を提案した。特性ばらつきの増大により、特性ばらつきによるリーク電流増加と故障による電流増加を区別することが困難になっており、IDDQ テストにおける故障見逃し、歩留まり損失の大きな要因となっている。本手法では、まず、特性パラメータ推定手法結果から、チップ毎の統計的なリーク電流分布を計算した。これを基に IDDQ テストの良否判定基準を決定した。特性パラメータ推定手法として、第2章で提案した IDDQ シグネチャを用いた手法を適用した。しかし、デバイスパラメータ推定は、良否判定を行う前であるため、IDDQ シグネチャが故障の影響を含む可能性がある。故障があった場合、故障の影響により IDDQ シグネチャは変調しており、正しいデバイスパラメータ推定ができない。そこで、まず、測定した IDDQ シグネチャから故障の影響を除外した故障無し IDDQ シグネチャを計算した。その後、これを用いて第2章で提案した手法を用いてデバイスパラメータを推定した。ここで、故障無し IDDQ シグネチャの推定計算時間が長い課題がある。そのため、本 IDDQ テスト手法では、パラメータ推定に基づく良否判定基準の設定前に、クラスタリングアルゴリズムを適用して、大きなリーク電流を生じる故障を含むチップを不良と判定した。このように、2段階の IDDQ テストを行うことで、良否判定基準計算の対象となる DUT 数を削減し、全体の計算時間を削減した。提案手法は、リーク電流内の特性ばらつき成分と故障成分を分離することができるため、適切な判定基準を設定でき、従来の IDDQ テスト手法と比べてテスト品質を向上できた。計算機実験において、歩留まり損失が生じることなく故障見逃しを 1.38 % に抑制し、既存手法である NNR 法と比べてテスト品質を 14 倍改善できることを示した。また、提案手法を適用することで、ノミナル条件時の IDDQ 電流の 40 % の大きさのリーク電流故障に対しては、その 80 % 以上を検出できることを示した。

第4章では、デバイスパラメータ推定結果を用いた適応型パス遅延故障テストを提案した。特性ばらつきの増大によりトランジスタの素子遅延が変動し、結果としてパス遅延値がシステムクロック周期を超える故障が問題となる。これらの故障を低コストでテストできる技術が要求される。適応型パス遅延故障テストでは、特性ばらつき推定結果に応じて、テストするパスを適応的に変更する。設計段階でパスクラスタリングを適用して、デバイスパラメータ値に応じてクリティカルパス集合のまとめ上げを行った。続いて、各クリティカルパス集合に対して、パス遅延故障テストパターン生成を行った。

テスト段階では、デバイスパラメータ推定結果に基づいて、テストに用いるテストパターンを適応的に変更した。また、適応型パス遅延故障テストを行った場合のパラメトリック故障の検出率を定量的に評価するために、パラメトリック故障に対するテストパターンの故障検出率を提案した。この検出率は、テスト対象回路の全パスにおけるパラメトリック故障確率とテストされたパスにおけるパラメトリック故障発生確率の比として定義した。計算機実験にて、従来の固定的なパス遅延テストと同等のテスト品質を保持しつつ、使用されるテストパターン数の期待値を10分の1に削減し、テストコストを削減できることを示した。

本論文では、テスト品質向上とテストコスト削減を目的として、特性ばらつき推定に基づく適応型テストを提案した。提案手法を適用することで、従来手法と比べてIDDQテスト品質を14倍向上し、パス遅延故障テストのテストパターン数の期待値を10分の1以下に削減できることを示した。従来からDUT毎にテスト内容を変更する適応型テストの概念はあるが、テスト技術者の経験に依存したものが多い。本研究で提案した適応型テストでは、特性ばらつき推定に基づいて適応的にテスト内容を変更する点が新規であり、新たに検討すべき分野を切り開いたと言える。デバイスパラメータ推定では特性ばらつきの分布をモデル化し、ベイズ推定法、最尤推定法を適用した。適応的なIDDQテスト、パス遅延故障テストにおいては、事前にリーク電流値とパス遅延値の期待される範囲を統計的に見積り、推定したデバイスパラメータからDUT毎のリーク電流値とパス遅延値を推定してテスト内容を最適化した。計算機実験において、提案手法によるテスト品質の向上とテストコストの削減が確認でき、その有用性を示した。

今後、提案した適応型テストを実用化するために、適応型テスト手法の効果を実デバイスにて確認する必要がある。

5.2 今後の展望

適応型テストは、適応的にテスト内容を変更するため、テスト品質とテストコストを制御する手法としても利用できる。そこで、適応型テストを中心とした、様々な製品カテゴリに対応可能なLSIテストプラットフォームを開発する。一般に、製品カテゴリに応じて要求されるテスト品質は異なる[82]。例えば、携帯電話やパーソナルコンピュータのような製品サイクルの短い民生品の要求テスト品質は、50 DPPMから2,000 DPPMである。ここで、DPPMはDefective Parts Per Millionの略で、百万チップ当りの不良

チップ数を示す．一方で，車載 LSI などの人命に関わるカテゴリでは，ゼロディフェクト，つまり不良率 0 DPPM が要求される．民生用に販売されていた LSI を車載向けに販売する場合もあり，従来の固定的なテストは，効率的なテスト品質の向上，テストコストの削減が容易ではない．本テストプラットフォームの開発により，単一デザインのチップを多商品カテゴリへの展開を可能にする．さらに，本テストプラットフォームでは，直流特性テストと交流特性テストの測定結果，故障診断結果等のテストデータ，統計的パス遅延分布，レイアウト設計データ等の設計データ，製造レシピ等の製造データを追加する．これらをデータベース化して解析することで，潜在的な信頼性不良の早期発見，テスト項目の更なる最適化に繋がる余地がある．

また，本研究で提案した適応型テストは，デバイス特性を推定する点が大きな特徴である．エネルギー問題の解決に関しては，駆動電源電圧を極限まで低減した超低消費電力 LSI の実用化が必要不可欠である．過度に電源電圧を下げると，しきい値電圧のばらつきが相対的に大きくなるため，微小な特性ばらつきの影響によりパラメトリック故障が頻発する．そこで，本研究で提案した適応型テスト手法の超低電力 LSI への適用を検討し，応用と改善を行う．

これらの研究により，高信頼かつ低コストな LSI の柔軟な提供が可能となることが期待される．今後，スマート社会の実現に向けて，エネルギー，安全，医療などの社会問題を解決する手段として LSI が大きな役割を果たすと予想され，スマート社会の実現に向けて大いに寄与するものと考えられる．

参考文献

- [1] 藤原秀雄: コンピュータの設計とテスト, 工学図書株式会社 (1990).
- [2] Semiconductor Industry Association: International Technology Roadmap for Semiconductors, 2011 Edition.
- [3] Segura, J., Keshavarzi, A., Soden, J. and Hawkins, C.: Parametric failures in CMOS ICs - a defect-based analysis, *Proceedings of IEEE International Test Conference*, pp. 90–99 (2002).
- [4] Gupta, P. and Papadopoulou, E.: *The Handbook of Algorithms for VLSI Physical Design Automation*, Taylor & Francis CRC Press (2008).
- [5] Berglund, C. N.: A unified yield model incorporating both defect and parametric effects, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 9, pp. 447–454 (1996).
- [6] Rearick, J.: A Survey of Test Problems and Solutions, *Proceedings of IEEE International Test Conference*, pp. 1–10 (2006).
- [7] Sengupta, S., Lundu, S., Chakravarty, S., Parvathla, P., Galivanche, R., Kosonocky, G., Rodgers, M. and Mak, T.: Defect-Based Test: A Key Enabler for Successful Migration to Structural Test, *Intel Technical Journal*, pp. 1–14 (1999).
- [8] Carbine, A. and Feltham, D.: Pentium Pro processor design for test and debug, *IEEE Design & Test of Computers*, Vol. 15, No. 3, pp. 77–82 (1998).
- [9] Sabade, S. S. and Walker, D. M.: IC Outlier Identification Using Multiple Test Metrics, *IEEE Design & Test of Computers*, Vol. 22, No. 6, pp. 586–595 (2005).

- [10] Nahar, A., Butler, K. M., Jr., J. M. C. and Weinberger, C.: Quality Improvement and Cost Reduction Using Statistical Outlier Methods, *Proceedings of International Conference on Computer Design*, pp. 64–69 (2009).
- [11] Nahar, A. and Daasch, R.: Burn-in Reduction using Principal Component Analysis, *Proceedings of International Conference on Computer Design*, pp. 1–10 (2005).
- [12] 米田 友洋, 土屋 達弘, 梶原 誠治: ディペンダブルシステム–高信頼システム実現のための耐故障・検証・テスト技術, 共立出版 (2005).
- [13] Borkar, S.: Designing reliable systems from unreliable components: the challenges of transistor variability and degradation, *Proceedings of IEEE/ACM International Symposium on Microarchitecture*, Vol. 25, No. 6, pp. 10–16 (2005).
- [14] Chakravaty, S. and Thadikaran, P. J.: *Introduction to IDDQ Testing*, Kluwer Academic Publisher (1997).
- [15] Sabade, S. S. and Walker, D.: IDDQ test: will it survive the DSM challenge?, *IEEE Design & Test of Computers*, Vol. 19, No. 5, pp. 8–16 (2002).
- [16] Mukhopadhyay, S., Bhunia, S. and Roy, K.: Modeling and analysis of loading effect on leakage of nanoscaled bulk-CMOS logic circuits, *IEEE Transactions on Industrial Electronics*, Vol. 25, pp. 1486–1495 (2006).
- [17] University of California, Berkeley: BSIM4v4.7, University of California (2011). [Online]. Available: <http://www-device.eecs.berkeley.edu/bsim/Files/BSIM4/BSIM470>.
- [18] Visweswariah, C., Ravindran, K., Kalafala, K., Walker, S. G., Narayan, S., Beece, D. K., Piaget, J., Venkateswaran, N. and Hemmett, J. G.: First-order incremental block-based statistical timing analysis, *Proceedings of IEEE/ACM Design Automation Conference*, pp. 331–336 (2004).

- [19] 平本俊郎, 竹内潔, 西田彰男: MOS トランジスタのスケーリングに伴う特性ばらつき, 電子情報通信学会誌, Vol. 92, No. 6, pp. 416–426 (2009).
- [20] Mitra, S., Volkerink, E., McCluskey, E. and Eichenberger, S.: Delay defect screening using process monitor structures, *Proceedings of IEEE VLSI Test Symposium*, pp. 43–48 (2004).
- [21] Smith, G. L.: Model for delay faults based upon paths, *Proceedings of IEEE International Test Conference*, pp. 342–349 (1985).
- [22] Sivaraman, M. and Strojwas, A. J.: Delay fault coverage: a realistic metric and an estimation technique for distributed path delay faults, *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 494–501 (1996).
- [23] Krstic, A. and Cheng, K.-T.(eds.): *Delay fault testing for VLSI circuits*, Kluwer Academic Publishers (1998).
- [24] 梶原誠司, 佐藤康夫: 論理回路に対する遅延テスト手法, 電子情報通信学会誌基礎・境界ソサイエティ Fundamentals Review, Vol. 1, No. 3, pp. 71–77 (2008).
- [25] Sato, Y., Hamada, S., Maeda, T., Takatori, A., Nozuyama, Y. and Kajihara, S.: Invisible Delay Quality - SDQM Model Lights Up What Could Not Be Seen, *Proceedings of IEEE International Test Conference*, pp. 1202–1210 (2005).
- [26] Gattiker, A. E. and Maly, W.: Current Signatures, *Proceedings of IEEE VLSI Test Symposium*, pp. 112–117 (1996).
- [27] Thibeault, C.: Improving Delta-IDDQ-based test methods, *Proceedings of IEEE International Test Conference*, pp. 207–216 (2000).
- [28] Powell, T., Pair, J., St.John, M. and Counce, D.: Delta IDDQ for Testing Reliability, *Proceedings of IEEE VLSI Test Symposium*, pp. 439–443 (2000).
- [29] Maxwell, P., O'Neill, P., Aitken, R., Dudley, R., Jaarsma, N., Quach, M. and Wiseman, D. M.: Current Ratios: A Self-Scaling Technique

- for Prodction IDDQ Testing, *Proceedings of IEEE International Test Conference*, pp. 738–746 (2000).
- [30] Rao, R. R., Devgan, A., Blaauw, D. and Sylvester, D.: Parametric Yield Estimation Considering Leakage Variability, *Proceedings of IEEE/ACM Design Automation Conference*, pp. 442–447 (2004).
- [31] Rao, R. R., Blaauw, D., Sylvester, D. and Devgan, A.: Modeling and Analysis of Parametric Yield under Power and Performance Constraints, *IEEE Design & Test of Computers*, Vol. 22, No. 4, pp. 376–385 (2005).
- [32] Li, X., Le, J. and Pileggi, L. T.: Projection-Based Statistical Analysis of Full-Chip Leakage Power with Non-Log-Normal Distributions, *Proceedings of IEEE/ACM Design Automation Conference* (2006).
- [33] Daasch, W. R., Cota, K., McNames, J. and Madge, R.: Neighbor Selection for Variance Reduction in IDDQ and Other Parametric Data, *Proceedings of IEEE International Test Conference*, pp. 92–100 (2001).
- [34] Turakhia, R., Benware, B. and Madge, R.: Defect Screening Using Independent Component Analysis on IDDQ, *Proceedings of IEEE VLSI Test Symposium*, pp. 427–432 (2005).
- [35] Padmanaban, S. and Tragoudas, S.: A critical path selection method for delay testing, *Proceedings of IEEE International Test Conference*, pp. 232–240 (2004).
- [36] Callegari, N., Bastani, P., Wang, L.-C., Chakravarty, S. and Tetelbaum, A.: Path selection for monitoring unexpected systematic timing effects, *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 781–786 (2009).
- [37] Iyengar, V., Xiong, J., Venkatesan, S., Zolotov, V., Lackey, D., Habitz, P. and Visweswariah, C.: Variation-aware performance verification using at-speed structural test and statistical timing, *Proceedings of IEEE International Test Conference*, pp. 405–412 (2005).
- [38] Chung, J. and Abraham, J. A.: Recursive path selection for delay fault testing, *Proceedings of IEEE VLSI Test Symposium*, pp. 65–70 (2009).

- [39] Liou, J.-J., Cheng, K.-T. and Mukherjee, D. A.: Path selection for delay testing of deep sub-micron devices using statistical performance sensitivity analysis, *Proceedings of IEEE VLSI Test Symposium*, pp. 97–104 (2000).
- [40] Eggersgluß, S., Fey, G., Glowatz, A., Hapke, F., Schloeffel, J. and Drechsler, R.: MONSOON: SAT-based ATPG for Path Delay Faults Using Multiple-Valued Logics, *Journal of Electronic Testing: Theory and Applications*, Vol. 26, No. 3, pp. 307–322 (2010).
- [41] Zolotov, V., Xiong, J., Fatemi, H. and Visweswariah, C.: Statistical path selection for at-speed test, *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pp. 624–631 (2008).
- [42] S. Natarajan, A. Krishnamachary, E. C. and Galivanche, R.: Path coverage based functional test generation for processor marginality validation, *Proceedings of IEEE International Test Conference*, pp. 1–9 (2010).
- [43] Cory, B. D. and Underwood, R. K. B.: Speed binning with path delay test in 150-nm technology, *IEEE Design & Test of Computers*, Vol. 20, No. 5, pp. 41–45 (2003).
- [44] : OpenCores. [Online]. Available: <http://www.opencores.org>.
- [45] Madge, R., Benware, B. R. and Daasch, W. R.: Obtaining high defect coverage for frequency-dependent defects in complex ASICs, *IEEE Design & Test of Computers*, Vol. 20, No. 5, pp. 46–53 (2003).
- [46] Pang, L.-T. and Nikolic, B.: Impact of Layout on 90nm CMOS Process Parameter Fluctuations, *VLSI Circuits, Digest of Technical Papers*, pp. 69–70 (2006).
- [47] Takahashi, T., Uezono, T., Shintani, M., Masu, K. and Sato, T.: On-die parameter extraction from path-delay measurements, *Proceedings of IEEE Asian Solid-State Circuits Conference*, pp. 101–104 (2009).
- [48] Shinkai, K. and Hashimoto, M.: Device-Parameter Estimation with On-chip Variation Sensors Considering Random Variability, *Proceed-*

- ings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 683–688 (2011).
- [49] Mahfuzul, I. A. K. M., Tsuchiya, A., Kobayashi, K. and Onodera, H.: Variation-Sensitive Monitor Circuits for Estimation of Global Process Parameter Variation, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 25, No. 4, pp. 571–580 (2012).
- [50] Uezono, T., Takahashi, T., Shintani, M., Hatayama, K., Masu, K., Ochi, H. and Sato, T.: Scan based process parameter estimation through path-delay inequalities, *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 3553–3556 (2010).
- [51] Rao, R., Srivastava, A., Blaauw, D. and Sylvester, D.: Statistical analysis of subthreshold leakage current for VLSI circuits, *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, Vol. 12, No. 2, pp. 131–139 (2004).
- [52] Brglez, F., Bryan, D. and Koiminski, K.: Combinational profiles of sequential benchmark circuits, *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1929–1934 (1989).
- [53] Synopsys, Inc.: *TetraMAX ATPG User Guide Version D-2010.06* (2010).
- [54] Synopsys, Inc.: *HSPICE User Guide: Basic Simulation and Analysis Version D-2010.03* (2010).
- [55] Pelgrom, M. J. M., Duinmayer, A. C. J. and Welbers, A. P. G.: Matching Properties of MOS Transistors, *IEEE Journal of Solid-State Circuits*, Vol. 24, No. 5, pp. 1433–1440 (1989).
- [56] Daasch, W., McNames, J., Bockelman, D., Cota, K. and Madge, O.: Variance Reduction using Wafer Patterns in IDDQ Data, *Proceedings of IEEE International Test Conference*, pp. 189–198 (2000).
- [57] Lin, C. J. and Reddy, S. M.: On Delay Fault Testing in Logic Circuits, *IEEE Transactions on Industrial Electronics*, Vol. 6, pp. 694–703 (1987).
- [58] Synopsys, Inc.: *PrimeTime Fundamental User Guide Version D-2010.06* (2010).

- [59] Saxena, S., Hess, C., Karbasi, H., Rossoni, A., Tonello, S., McNamara, P., Lucherini, S., Minehane, S., Dolainsky, C. and Quarantelli, M.: Variation in Transistor Performance and Leakage in Nanometer-Scale Technologies, *IEEE Transactions on Electron Devices*, Vol. 55, No. 1, pp. pp131–pp144 (2008).
- [60] Kaeriyama, S., Kajita, M. and Mizuno, M.: A 1-to-2GHz 4-Phase On-Chip Clock Generator with Timing-Margin Test Capability, *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 175–594 (2007).
- [61] R Development Core Team: R. [Online]. Available: <http://www.r-project.org>.
- [62] Jandhyala, S., Balachandran, H. and Jayasumana, A.: Clustering-based Techniques for IDDQ Testing, *Proceedings of IEEE International Test Conference*, pp. 724–729 (1999).
- [63] Jandhyala, S., Balachandran, H. and Jayasumana, A.: Clustering Based Evaluation of IDDQ Measurements: Applications in Testing and Classification of ICs, *Proceedings of IEEE VLSI Test Symposium*, pp. 444–449 (2000).
- [64] MacQueen, J.: Some methods for classification and analysis of multivariate observations, *5-th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967).
- [65] Nakamura, Y. and Tanaka, M.: A Multi-dimensional Iddq Testing Method Using Mahalanobis Distance, *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 303–309 (2010).
- [66] Yilmaz, E., Ozev, S. and Butler, K. M.: Adaptive multidimensional outlier analysis for analog and mixed signal circuits, *Proceedings of IEEE International Test Conference*, pp. 1–8 (2011).
- [67] Madge, R., Benware, B., Ward, M. and Daasch, R.: The value of statistical testing for quality, yield and test cost improvement, *Proceedings of IEEE International Test Conference*, pp. 322–332 (2005).

- [68] Kirkpatrick, S., Jr., C. D. G. and Vecchi, M. P.: Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598, pp. 671–680 (1983).
- [69] Rousseeuw, P. J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis, *Journal of Computational & Applied Mathematics*, pp. 53–65 (1987).
- [70] Xiong, J., Shi, Y., Zolotov, V. and Visweswariah, C.: Statistical multilayer process space coverage for at-speed test, *Proceedings of IEEE International Test Conference*, pp. 1–9 (2009).
- [71] Benner, S. and Boroffice, O.: Optimal Production Test Times Through Adaptive Test Programming, *Proceedings of IEEE International Test Conference*, Los Alamitos, CA, USA, IEEE Computer Society, pp. 908–915 (2001).
- [72] Gotkhindikar, K. R., Daasch, W. R., Bulter, K. M., Jr., J. M. C. and Nahar, A.: Die-level adaptive test: Real-time test reordering and elimination, *Proceedings of IEEE International Test Conference*, pp. 1–10 (2011).
- [73] Sumikawa, N., Drmanac, D. G., Wang, L.-C., Winemberg, L. and Abadir, M. S.: Forward prediction based on wafer sort data – A case study, *Proceedings of IEEE International Test Conference*, pp. 1–10 (2011).
- [74] Uezono, T., Takahashi, T., Shintani, M., Hatayama, K., Masu, K., Ochi, H. and Sato, T.: Path clustering for adaptive test, *Proceedings of IEEE VLSI Test Symposium*, pp. 15–20 (2010).
- [75] Chan, T.-B. and Kahng, A. B.: Improved Path Clustering for Adaptive Path-Delay Testing, *Proceedings of IEEE International Symposium on Quality Electronic Design*, pp. 13–20 (2012).
- [76] Cheng, K.-T. and Chen, H.-C.: Classification and identification of nonrobust untestable path delay faults, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 8, pp. 845–853 (1996).
- [77] Yoshikawa, Y., Ohtake, S., Inoue, T. and Fujiwara, H.: Fast False Path Identification Based on Functional Unsensitizability Using RTL

- Information, *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 660–665 (2009).
- [78] Arslan, B. and Orailoglu, A.: Full Exploitation of Process Variation Space for Continuous Delivery of Optimal Delay Test Quality, *Proceedings of IEEE/ACM Asia and South Pacific Design Automation Conference*, pp. 552–557 (2013).
- [79] Synopsys, Inc.: *PrimeTime VX User Guide version D-2010.06* (2010).
- [80] Synopsys, Inc.: *Design Compiler User Guide version D-2010.06* (2010).
- [81] Arthur, D. and Vassilvitskii, S.: k-means++: The Advantages of Careful Seeding, *Proceedings of ACM-SIAM Symposium on Discrete Algorithms*, pp. 1027–1035 (2007).
- [82] Semiconductor Industry Association: *International Technology Roadmap for Semiconductors*, 2013 Edition.

付録

SLL の精度は Monte Carlo 回路シミュレーションにおける標本数によって決定される．ここでは，Monte Carlo 回路シミュレーションの試行回数と要求されるパラメータ推定精度の関係について述べる．

Monte Carlo 回路シミュレーションにおいて，正規分布から N 回の標本数を取得した場合を考える．取得した標本値を対数化することで，リーク電流と考えることができる．ここで，未知母数の平均と分散をそれぞれ μ と σ^2 とする．

取得した標本 \bar{x} は正規分布に従うため， $\bar{x} \sim N(\mu, \sigma^2/N)$ と表せる． \bar{x} に対して 95 % の信頼区間が要求された場合， \bar{x} は次式を満たす必要がある．

$$\mu - 1.96 \frac{\sigma}{\sqrt{N}} < \bar{x} < \mu + 1.96 \frac{\sigma}{\sqrt{N}} \quad (1)$$

未知分散 σ^2 に不偏分散 u^2 を代入することで，式 (1) は式 (2) のようになる．

$$-1.96 \frac{u}{\sqrt{N}} \leq \bar{x} - \mu \leq 1.96 \frac{u}{\sqrt{N}} \quad (2)$$

式 (2) は真値 $\bar{x} - \mu$ との推定誤差を表す．図 1 に，65 nm プロセスのインバータ素子を用いた場合の N に対する 95 % 信頼性区間の幅の変化を示す．図 1 から，標本数 N を増やすことで，信頼性区間を小さくできていることが分かる． $N = 1000$ の時，95 % 信頼性区間の幅は， $|\bar{x} - \mu| \leq 0.0907 \log A$ となる．

続いて，標本分散 s^2 について考える． s^2 と σ^2 の相対誤差は，自由度 $N-1$ の Ns^2/σ^2 がカイ二乗分布に従うという特徴を用いることで評価できる．例えば，95 % の信頼区間が要求された場合，次式のようになる．

$$\frac{X_{N-1}(0.025)}{N} \leq \frac{s^2}{\sigma^2} \leq \frac{X_{N-1}(0.975)}{N} \quad (3)$$

ここで， X_{N-1} は自由度 $N-1$ のカイ二乗分布の累積分布関数である．図 2 は，標本数 N に対する s^2/σ^2 の 95 % の信頼区間幅の推移を示す．標本数 $N = 1000$ の時，95 % の信頼区間の幅は $0.9142 \leq s^2/\sigma^2 \leq 1.0895$ となる．

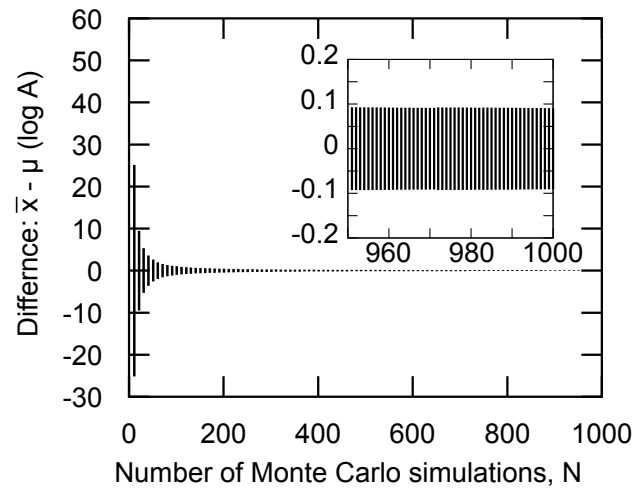


図 1: 標本数 N に対する $\bar{x} - \mu$ の 95%信頼性区間幅の推移

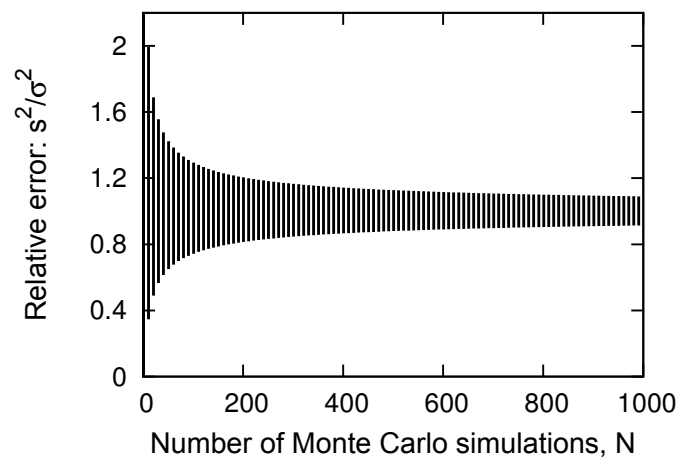


図 2: 標本数 N に対する s^2/σ^2 の 95%信頼性区間幅の推移

図目次

1.1	ITRS ロードマップによるパラメトリック故障発生確率予測 [2]	2
1.2	IDDQ テスト良否判定基準決定の困難化	6
1.3	パラメトリック故障の概念図	7
1.4	パス遅延故障モデル	8
1.5	IDDQ 電流シグネチャの例	9
1.6	一般的なパス遅延故障テスト適用フロー	10
1.7	想定するばらつき空間におけるクリティカルパス数の変化	11
1.8	推定に基づく適応型テストの概念	12
1.9	適応型パス遅延故障テストの概念図	13
1.10	本論文で想定するテストの流れと論文の構成	14
2.1	IDDQ シグネチャを用いたデバイスパラメータ推定手法の全体フロー	19
2.2	2 ばらつき変数によるプロセス領域分割例	20
2.3	2 入力 NAND ゲートのゲートレベル SLL の例. 本ライブラリは, 全パラメータ条件, 全入力状態におけるリーク電流の確率密度分布を格納する.	21
2.4	統計的リーク電流ライブラリ生成工程の処理	22
2.5	統計的リーク電流分布計算工程の処理	23
2.6	ベイズの定理に基づくデバイスパラメータ推定工程の処理	24
2.7	尤度の概念図	25
2.8	小領域毎のテストパターン $t = 1$ の時の s38584 のリーク電流マップ	27
2.9	各小領域における, テストパターン $t = 1$ の時の全リーク電流の対数正規分布の μ と σ	28
2.10	(-10 mV, -10 mV) における IDDQ シグネチャ	29
2.11	(-10 mV, -10 mV) に対する, 最後のテストパターンの時のデバイスパラメータ推定結果	30

2.12	テストパターン $t = 3$ の時の $(-10\text{ mV}, -10\text{ mV})$ に対するデバイスパラメータ推定結果	30
2.13	正解候補 5 領域における, テストパターン t に対する推定確率 $P(x I_t)$ の推移	31
2.14	$(10\text{ mV}, 10\text{ mV})$ と $(-10\text{ mV}, -10\text{ mV})$ の IDDQ シグネチャ	31
2.15	$(10\text{ mV}, 0\text{ mV})$ における, 最終テストパタンの時のデバイスパラメータ推定結果	32
2.16	$(-10\text{ mV}, -10\text{ mV})$ と $(10\text{ mV}, 0\text{ mV})$ におけるリーク電流の確率密度分布	33
2.17	最終テストパターンにおけるの全小領域のデバイスパラメータ推定結果	34
2.18	$(70\text{ mV}, 80\text{ mV})$ と $(80\text{ mV}, 80\text{ mV})$ における, テストパターン t に対する推定確率 $P(x I_t)$ の推移	35
2.19	F_{\max} テストにおける異常値選別手法	38
2.20	2 チップにおける F_{\max} シグネチャの例	39
2.21	F_{\max} テストの枠組みを用いたデバイスパラメータ手法の全体フロー	40
2.22	SMPDL 生成の手続き	42
2.23	10 点に対する ΔV_{thn} と ΔV_{thp} の推定結果	47
2.24	DBE 方式推定と MLE 方式推定におけるテストパターンに対する ΔV_{thn} と ΔV_{thp} の推定推移	48
2.25	1,000 点に対する ΔV_{thn} と ΔV_{thp} の推定誤差のヒストグラム	49
3.1	2 段階 IDDQ テスト手法の概念図	55
3.2	クラスタリング方式フィルタの概念図	56
3.3	デバイスパラメータ推定に基づく IDDQ テスト良否判定基準決定の全体フロー	58
3.4	IDDQ 電流値の統計的リーク電流分布に対する当てはまりの良さの概念図	60
3.5	最適化関数計算の処理	61
3.6	デバイスパラメータと局所ばらつきを考慮した統計的リーク電流分布の計算方法	63
3.7	仮想ウェハ上のデバイスパラメータの分布	65
3.8	仮想ウェハ上の $k = 1$ と $k = 20$ の時の IDDQ 電流分布	65

3.9	(a) 推定値計算のためのテンプレート. DUT の周辺 8 チップ を推定値計算に使用する (b) 289 DUT に対する測定 IDDQ シ グネチャの平均値と残差のヒストグラム.	66
3.10	k 平均法によって分離された C_L と C_H のヒストグラムの例. この例においては, 2 つのヒストグラムが明確に分離されて いる.	68
3.11	図 3.10 におけるシルエットプロットの例	68
3.12	故障無し IDDQ シグネチャの推定例	69
3.13	2 模擬チップにおける IDDQ 良否判定基準の例. 一方は良品 チップで, もう一方は不良品チップである.	70
3.14	リーク故障サイズと故障見逃しの関係	71
4.1	適応型パス遅延テストフロー	76
4.2	各小領域におけるテスト対象パス集合 [74].	77
4.3	文献 [74] におけるパスクラスタリング結果 (d_x , d_y は共に 5 mV). 同じ記号は同じクラスタに属することを示す. . . .	78
4.4	テストフローチャート	80
4.5	pfc の概念図	82
4.6	$(\Delta V_{thn}, \Delta V_{thp}) = (80 \text{ mV}, -40 \text{ mV})$ の時の, テストパターン数に 対する pfc' の関係	89
4.7	クラスタ数 3 の時のパスクラスタリング結果	90
4.8	クラスタ数 5 の時のパスクラスタリング結果	91
4.9	クラスタ数 10 の時のパスクラスタリング結果	91
4.10	各パスクラスタに対するパス遅延テストパターン数とパス遅延 故障検出率	92
1	標本数 N に対する $\bar{x} - \mu$ の 95 % 信頼性区間幅の推移	110
2	標本数 N に対する s^2/σ^2 の 95 % 信頼性区間幅の推移	110

表 目 次

2.1	各工程における CPU 時間	29
2.2	推定する ΔV_{thn} と ΔV_{thp} の組み合わせ (mV)	46
3.1	クラスタリング方式フィルタ結果	67
3.2	異なる良否判定基準に対する歩留まり損失と故障見逃し	69
4.1	STARC03 仕様 [25].	84
4.2	従来手法 (conv-nom, conv-worst) におけるパス遅延 ATPG 結果	85
4.3	適応型テスト (ours) におけるパス遅延 ATPG 結果	85
4.4	従来手法と提案手法の PFC 比較	86
4.5	ΔV_{thp} と ΔV_{thn} の同時確率	86
4.6	従来手法 conv-nom における小領域毎の pfc'	87
4.7	従来手法 conv-worst における小領域毎の pfc'	87
4.8	提案手法 ours における小領域毎の pfc'	88
4.9	パスクラスタリングを用いた適応型テスト結果	93

謝辞

本研究の機会を与えて頂き、研究のご指導を賜りました京都大学大学院情報学研究科 佐藤高史教授に心より感謝いたします。2011年に社会人博士として入学し、佐藤先生のもとで本研究を進めて参りました。業務後の深夜まで及ぶ個別指導や、メール、電話会議でもご指導いただき、社会人として業務と学業を両立しながら研究を進めることができました。常に熱心な指導と暖かい叱咤激励を賜りました。心から御礼申し上げます。同研究科小野寺秀俊教授、高木直史教授には本論文をまとめるにあたって貴重な助言をいただいたことに深く感謝します。同研究科 越智裕之准教授(現在 立命館大学教授)には、研究について適切な御助言や励ましの御言葉を頂いたことに深く感謝いたします。また、同研究科 筒井弘助教(現在 北海道大学准教授)には、計算機環境の面で助言いただくとともに、有益な御助言や励ましの御言葉を頂き深く感謝致します。同研究科 廣本正之助教には、論文執筆や研究の進め方について熱心にご指導いただき深く感謝いたします。東京工業大学 益一哉教授には研究当初の方針についてご助言頂きました。ここに深く感謝致します。

また、本研究を進めるにあたり、株式会社半導体理工学研究センター 開発第2部 テスト&故障解析開発室 相京隆博士(現在 同社研究開発部部長)と 畠山一実博士(現在 群馬大学協力研究員)に、研究開発の推進及び内容を深く御討論頂きました。ここに深く感謝致します。また東京工業大学 益研究室の上菌巧博士(現在 日立製作所)と高橋知之氏(現在 東芝)には深い討論、助力をいただき深く感謝致します。

本研究につきまして、佐藤高史研究室の大学院生、学部生、秘書の皆様には有益なご助言、ご助力をいただきました。ここに深く感謝致します。

本研究は東京大学大規模集積システム設計教育研究センターを通して行われたものである。同センターの皆様及び関係者の皆様に感謝致します。

本論文は、父 新谷辰男と母 新谷京子が支えてくれたことで完成させることができた。最後に、いつも支えてくれた家族に感謝します。

著者による発表論文

学術論文

1. Michihiro Shintani and Takashi Sato: “Device-Parameter Estimation Through IDDQ Signatures,” IEICE Transactions on Information and Systems, Vol. E96-D, No. 2, pp. 303–313, Feb. 2013.
2. Michihiro Shintani, Takumi Uezono, Tomoyuki Takahashi, Kazumi Hatayama, Takashi Aikyo, Kazuya Masu, and Takashi Sato: “A Variability-Aware Adaptive Test Flow for Test Quality Improvement,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 33, Issue 7, pp. 1056–1066, Jul. 2014.
3. Michihiro Shintani and Takashi Sato: “IDDQ Outlier Screening through Two-phase Approach: Clustering-based Filtering and Estimation-based Current-threshold Determination,” IEICE Transactions on Information and Systems, Vol. E97-D, No. 8, pp. 2095–2104, Aug. 2014.

査読付き会議

1. Tomoyuki Takahashi, Takumi Uezono, Michihiro Shintani, Kazuya Masu, and Takashi Sato: “On-die parameter extraction from path-delay measurements,” in Proc. of IEEE Asian Solid-State Circuits Conference, pp. 101–104, Nov. 2009.
2. Michihiro Shintani, Takumi Uezono, Tomoyuki Takahashi, Hiroyuki Ueyama, Takashi Sato, Kazumi Hatayama, Takashi Aikyo, and Kazuya Masu: “An Adaptive Test for Parametric Faults Based on Statistical Timing Information,” in Proc. of Asian Test Symposium, pp. 151–156, Nov. 2009.

3. Takumi Uezono, Tomoyuki Takahashi, Michihiro Shintani, Kazumi Hatayama, Kazuya Masu, Hiroyuki Ochi, and Takashi Sato: "Scan based process parameter estimation through path-delay inequalities," in Proc. of IEEE International Symposium on Circuits, and Systems, pp. 3553–3556, Mar. 2010.
4. Takumi Uezono, Tomoyuki Takahashi, Michihiro Shintani, Kazumi Hatayama, Kazuya Masu, Hiroyuki Ochi, and Takashi Sato: "Path clustering for adaptive test," in Proc. of VLSI Test Symposium, pp. 15–20, Apr. 2010.
5. Michihiro Shintani and Takashi Sato: "Getting the Most Out of IDDQ Testing," in Proc. of IEEE/ACM Workshop on Variability and Characterization, Nov. 2011.
6. Michihiro Shintani and Takashi Sato: "A Bayesian-Based Process Parameter Estimation Using IDDQ Current Signature," in Proc. of IEEE VLSI Test Symposium, pp. 86–91, Oct. 2012.
7. Michihiro Shintani and Takashi Sato: "Adaptive Current-Threshold Determination for Accurate IDDQ Testing," in Proc. of IEEE/ACM Workshop on Variability and Characterization Workshop, Nov. 2012.
8. Michihiro Shintani and Takashi Sato: "An Adaptive Current-Threshold Determination for IDDQ Testing Based on Bayesian Process Parameter Estimation," in Proc. of IEEE/ACM Asia and South Pacific Design Automation Conference, pp. 614–619, Jan. 2013 [IEICE VLD Excellent Student Author Award for ASP-DAC 2013].
9. Michihiro Shintani and Takashi Sato: "Sensorless estimation of global device-parameters based on F_{\max} testing," in Proc. of IEEE/ACM International Conference on Computer-Aided Design, accepted for presentation, 2014.

研究会及び全国大会

1. 新谷 道広, 高橋 知之, 植山 寛行, 上蘭 巧, 畠山 一実, 佐藤高史, 相京 隆, 益一哉: "統計的タイミング情報に基づく適応型テスト", 電子情報通信

- 学会総合大会, D-10-16, Mar. 2009.
2. 上蘭 巧, 高橋 知之, 植山 寛行, 新谷 道広, 佐藤高史, 相京 隆, 益一哉: “適応型テストにおけるクリティカルパスのクラスタリング手法”, 電子情報通信学会総合大会, D-10-17, Mar. 2009.
 3. 新谷 道広, 佐藤 高史: “プロセスばらつき推定に基づく IDDQ テスト良品判定基準決定の試み”, 電子情報通信学会技術研究報告, DC2011-84, pp. 49–54, Feb. 2012.
 4. 新谷 道広, 佐藤 高史: “IDDQ 電流による大域プロセスばらつきの推定手法”, 電子情報通信学会技術研究報告, VLD2011-120, pp. 1–6, Mar. 2012.
 5. 新谷 道広, 佐藤 高史: “パラメータ推定に基づく IDDQ 電流しきい値決定のオンラインテストに向けた高速化”, 電子情報通信学会技術研究報告, VLD2012-137, pp. 7–12, Mar. 2013.
 6. Michihiro Shintani and Takashi Sato: “[記念講演] An Adaptive Current-Threshold Determination for IDDQ Testing Based on Bayesian Process Parameter Estimation”, 電子情報通信学会技術研究報告, VLD2012-152, pp. 91–91, Mar. 2013.
 7. 新谷 道広, 佐藤 高史: “最大動作周波数テストの枠組みを用いたデバイスパラメータ推定手法”, 電子情報通信学会技術研究報告, DC2013-85, pp. 37–42, Feb. 2014.

表彰等

1. IEICE VLD Excellent Student Author Award for ASP-DAC 2013, 平成 25 年 3 月.